



HAL
open science

Localisation spatiale par subdivision pour l'accélération des calculs en radiométrie :

Jean-Christophe Roche

► **To cite this version:**

Jean-Christophe Roche. Localisation spatiale par subdivision pour l'accélération des calculs en radiométrie :. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 2000. Français. NNT : . tel-00006752

HAL Id: tel-00006752

<https://theses.hal.science/tel-00006752>

Submitted on 25 Aug 2004

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

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



THÈSE

Présentée par
Jean-Christophe ROCHE

Pour obtenir le grade de
Docteur de l'Université Joseph Fourier
(arrêtés ministériels du 5 juillet 1984 et du 30 mars 1992)

Discipline : **Mathématiques Appliquées**

Localisation spatiale par subdivision pour l'accélération des calculs en radiométrie : étude théorique et applications industrielles

Soutenue publiquement le 11 septembre 2000 devant le jury :

M. Marc DANIEL	<i>Président et Rapporteur</i>
M. Michel MÉRIAUX	<i>Rapporteur</i>
M. Christophe RABUT	<i>Rapporteur</i>
M. Bernard LACOLLE	<i>Examineur</i>
M. Patrick CHENIN	<i>Directeur de thèse</i>
M. Jacques DELACOUR	<i>Invité</i>

À ma fille,
à ma femme,
à mes parents,
à mes amis.

"Nous saurions beaucoup de choses, si nous savions ce qu'est un rayon lumineux."

Louis de Broglie

Avant-propos

Je tiens tout d'abord à exprimer ma profonde reconnaissance à Monsieur Patrick Chenin, maître de conférences à l'université Joseph Fourier, pour avoir dirigé cette thèse. Sa vision profonde et son expérience des différents domaines abordés au cours de ce projet ont été une source essentielle d'idées et d'encouragements. Je le remercie, en particulier, pour la lecture experte de mon mémoire.

Je remercie Monsieur Marc Daniel, professeur à l'École Supérieure d'Ingénieurs de Luminy, pour ses critiques constructives qui ont contribué à l'amélioration de ce mémoire. Nos nombreux échanges électroniques ainsi que son intérêt à l'égard de ce travail m'ont apporté beaucoup d'encouragements. Je le remercie, en particulier, pour avoir accepté de rapporter sur cette thèse. Il a aussi accepté de présider ce jury. Je le remercie chaleureusement de l'honneur qui m'est ainsi fait.

J'adresse également mes remerciements à Monsieur Michel Mériaux, professeur à l'université de Poitiers, et à Monsieur Christophe Rabut, professeur à l'Institut National des Sciences Appliquées de Toulouse, pour avoir accepté de rapporter sur cette thèse ainsi que pour leurs remarques précieuses.

Monsieur Bernard Lacolle, professeur à l'université Joseph Fourier, a accepté d'examiner mon mémoire. Je le remercie.

Je remercie enfin Monsieur Jacques Delacour, ingénieur opticien et Président Directeur Général de la société Optis, d'avoir accepté de participer à ce jury. Ses qualités scientifiques et humaines ont été un atout indispensable à la réussite d'un tel projet. Sa générosité, sa gentillesse et la confiance qu'il m'a cédée d'emblée ont rendu ces années très agréables et détendues. Ces remerciements sont à la hauteur de l'estime et de l'admiration que j'ai pour lui. Néanmoins, je regrette sincèrement que Monsieur Jacques Delacour n'ai pu participer au jury, ni faire représenter sa société.

Ce travail doit beaucoup aux échanges d'idées et aux heures passées avec de nombreuses personnes et en particulier Hafsa Deddi, Mohammed Khachan et Khaled Melkemi.

Une bonne ambiance au sein de la société Optis a constitué une source de motivation et d'inspiration. Je remercie très vivement Nathalie Altthaler, Hervé Bénard, Stéphane Bertou, Christophe Cariotis, François Commagnac, Thomas Demachy, François Doussoux, Joris Jourdain, Barthelemy Fondeur, Laurent Fournier, Sylvain Lecluze, Ludovic Manillier, Gilles Mathieu, Laurent Mayollet, Marc Mermet, Franck Nardin, Fabien Prugnard, Romain Ramel, Laurent Reignier, Nicolas Soudet et Jean-Claude Venturino.

Il en est de même pour le Laboratoire de Modélisation et Calcul, où, en particulier, les séminaires inter-thésards ont été un lieu de communication et d'échanges fructueux. Je remercie Guillaume Allègre, Loïc Benayoun, Ludovic Dumaine, Gerard Eichenmüller, Jean-Michel Fabre, Cédric Gerot, Alexandre Gerussi, Patrick Giroud, Miguel Juston, Voichita Maxim, Eckart Pfluegel, Catherine Pras, Luc Rebillard, Laurent Testard, Valérie Pham-Trong, Luc Pionchon ainsi que tous ceux que j'ai côtoyés durant ma thèse.

J'exprime toute ma reconnaissance et ma tendresse à ma famille et à mes amis, parce qu'ils sont tout simplement formidables.

Ma femme, Stéphanie, a fait preuve d'une grande patience. Son soutien moral ainsi que sa confiance permanente ont été des atouts indispensables pour mener à bien ce projet. Notre fille, Maëva, est plus qu'un rayon de soleil, c'est une joie de vivre quotidienne et un grand bonheur apporté par un enfant dont la soif de découvrir est immense.

Mes parents, Marie-France et Gilbert, m'ont apporté un soutien moral et financier indispensable à cette réussite. Notamment la dernière année de thèse, pour laquelle, sans eux, je n'aurais pu mener à terme cette enrichissante expérience de la vie professionnelle. Leur confiance accordée au cours de toutes ces années, et même les années précédentes, surtout dans les moments délicats, a été cruciale. Les mots ne suffisent pas à exprimer toute ma reconnaissance.

Ma sœur, Stéphanie, a semble-t-il été un peu stressée pour ma soutenance. C'est bien là une preuve d'encouragement. Elle découvre à son tour le difficile métier de parent. Hamid partage avec elle cette tâche pour le plus grand bonheur de la petite Léa.

Ma grand-mère, Paulette, nous a souvent concocter de bons petits plats typiques et naturels que nous avons dégustés ensemble. Quelle sensation de bien être quand je porte ses pulls en laine tricotés à la main. Et puis, une grand-mère, c'est irremplaçable.

Avec mes tantes et oncles, André, Anne-Marie, Elisabeth, Hervé, Jean-Pierre et Michel, j'ai passé d'excellents moments. Bien souvent les maxillaires ont grincé. Ces instantanés de la vie sont des souvenirs intarissables. Ils sont présents dans mon album mémoriel, pour lequel de nombreuses pages sont en attente des clichés à venir.

La vie est faite de hauts et de bas. Je ne retiendrai que les bons moments passés aux côtés de Roger. Et ceux-ci sont de plus en plus nombreux. Tant mieux. Il faut savoir profiter au maximum des bons moments, cela passe en partie par une vision au jour le jour de la vie. Mes pensées vont aussi à Marie-Françoise, Lotfi et leur fille Sarah.

Cécile, avec qui nous avons traversé un océan, dans les deux sens, et ainsi conquis une infime partie du territoire nord américain. Et tout cela pour aller manger, ou plutôt ingurgiter, des hamburgers. Heureusement, nous garderons un souvenir inoubliable de l'accueil très chaleureux réservé par les québécois. Sans oublier, dans d'autres instants, le millier de kilomètres franchit pour aller déguster des coquillages et crustacés ainsi que quelques dizaines de crêpes à la farine de blé noir. Bonjour les kilogrammes!

David, qui nous attends toujours de l'autre côté de la Manche, doit commencer à désespérer. C'est vrai, il nous faut encore un peu de temps pour se décider à traverser ce bras de mer, soit par dessus, soit par dessous. Il ne reste plus qu'à choisir. Au fait, j'ai encore le souvenir des genoux meurtris par une descente interminable. Heureusement qu'il y avait à l'arrivée une bonne raclette pour nous réchauffer. See you dans pas longtemps!

Keyvan, qui a goûté au plaisir de la vie au Québec, c'est non sans une certaine nostalgie qu'il a retrouvé le continent européen. Que de bons moments passés en sa compagnie. J'ai encore mal à la mâchoire quand je pense à certains éclats de rire, entre autres, jadis dans un amphithéâtre, lors de cours de thermodynamique. Et quand je pense aux kilogrammes de fromages qui sont passés par là. Et j'en passe. Au fait, c'est du Saint-Agur?

Olivier, dont les jambes sont programmées pour courir en toutes circonstances. C'est une vraie bible informatique ambulante, ainsi qu'un pâtissier averti. Certes, le mille-feuille en compte parfois un peu moins, mais avant tout l'important est le goût. Et ça rime! C'est aussi un cascadeur émérite, mais là je n'en dirai pas plus. Peut-être, un jour, verrons-nous sortir un site web, un livre, une chanson ou encore un film à sa gloire ...

Sylvain, dont la culture est un jardin dans lequel on découvre chaque jour de nouvelles variétés de toutes sortes. Comment oublier les quelques ballades en montagne, avec Jess, le plus fidèle des compagnons, allant même jusqu'à tenter une nocturne. Quoiqu'il en soit, le stress tombe dès que l'on passe à table. D'autant plus lorsqu'il s'agit d'exciter les papilles gustatives, notamment lors de voyages aux pays des vigneron. Avec en point de mire des objectifs de conquêtes dans des univers virtuels. C'est ici que les vraies difficultés commencent ...

Les nombreux moments de détente sont nécessaires à toute activité cérébrale, même minimale. Les nautiles-débarquent des quatre coins de la région. Armés d'un boîtier métallique pour la plupart, nous nous installons pour quelques minutes, voire parfois, mais rarement, quelques heures, à la poursuite du plus proche voisin. Dans quel but ? François nous le chanterait bien. Pourquoi ne pas envisager une chorale avec Éric, Jean-Michel, Jean-Roch, Jérôme, Serge et tous les autres ? Quoique, la météo était optimiste jusqu'à aujourd'hui ! Je pense signer mon arrêt de mort virtuelle en écrivant ces quelques lignes. Qu'à cela ne tienne, je suis avant tout un compétiteur. Alors à vos mulots ...

C'est avec Jean-Baptiste et Olivier que j'ai du nouer ma première cravate pour des motivations fortes, mais qui n'ont malheureusement par connues le succès attendu. L'essentiel est d'avoir essayé. Et puis, les expériences acquises sont des points positifs. Et encore plus les liens que nous avons conservés, même si l'on ne se voit pas très souvent.

Je dois beaucoup à mes anciens colocataires du bureau 56, Hafsa, Khaled et Mohammed. Que ce soit pour leur gentillesse, leur culture ou leurs nombreux conseils scientifiques, ils m'ont apportés bien plus que certaines personnes. Bien entendu, la plus grande joie est que ce sont des amis. La distance n'enlevant rien à ce sentiment. Qu'on se le dise : "in challah".

Mes souvenirs et pensées vont particulièrement à Brigitte, Christophe, Claude, Daniel, David, Florence, Gilles, Jean-Paul, Joël, Laurent, Lionel, Loïc, Serge, Stéphanie et tous les autres avec qui j'ai passé d'innombrables instants, que ce soit au bord de l'eau les doigts de pieds en éventail, dans les nombreuses soirées dansantes ou tout simplement ensemble. Je n'aurais pas assez de pages ici pour narrer tous ces moments. Peu importe, ce qui compte c'est la sensation de bonheur procurée. Pourvu que ça dure !

Des personnes du Ministère chargé de la Recherche, qui se reconnaîtrons à la lecture de ces quelques lignes, m'ont apportés un soutien concret et moral indispensable par rapport aux difficultés rencontrées au cours de ce projet. Qu'elles soient assurées de mes remerciements les plus sincères.

Le cycle de la vie est ainsi fait que certains proches ne sont plus là aujourd'hui. J'ai une pensée particulière et profonde à leur égard.

Résumé

Localisation spatiale par subdivision pour l'accélération des calculs en radiométrie : étude théorique et applications industrielles

La physique de la lumière ainsi que les outils géométriques pour la Conception Assistée par Ordinateur sont à la base des logiciels de simulation des phénomènes lumineux pour la fabrication des systèmes optiques. Ce n'est pas sans difficulté que les industriels conçoivent ces logiciels dont un des principaux handicaps est que les simulations sont très coûteuses en temps. L'objectif principal de ce travail est de rechercher et développer des algorithmes de calcul plus performants.

Dans un premier temps, on décrit précisément le modèle du transport des photons dans ce contexte, composé de l'équation de Boltzmann accompagné de conditions de bord, et qui, dans le cas de milieux homogènes par morceaux, se ramène à l'équation de radiosité.

Ensuite, on présente les outils géométriques utilisés dans le modéleur hybride CSG (*Constructive Solid Geometry*) et BRep (*Boundary Representation*) ainsi que les algorithmes de base nécessaires à la recherche d'intersections entre des demi-droites et des objets géométriques.

Puis, un tour d'horizon des méthodes d'accélération des calculs en radiométrie par localisation spatiale est présenté. En tenant compte des contraintes industrielles, une telle méthode d'accélération est alors adaptée au contexte puis développée dans un environnement logiciel existant. Des expérimentations numériques montrent l'efficacité des nouvelles bibliothèques.

Enfin, une étude théorique des complexités en temps et en mémoire liées aux méthodes de localisation spatiale, faisant intervenir les sommes de Minkowski d'ensembles géométriques, débouche sur une stratégie consistant à minimiser la complexité en temps pour choisir les paramètres de localisation.

Mots-clé : subdivision spatiale hiérarchique, complexités, équation de Boltzmann, radiométrie, lancer de rayons, modélisation géométrique, CAO, CSG, BRep, sommes de Minkowski

Abstract

Accelerated radiometric computation using subdivision for spatial localization : theoretical study and industrial applications

Physic of the light and geometrical tools for Computer Aided Geometric Design are the basis of light phenomenon simulation software for optical systems development. It is difficult for industrials to design and produce those software with a major handicap being the time needed for the simulations. The aim of this work is to search and to develop calculus algorithms with higher performances.

Firstly, we describe precisely the photon transport model in our context. We use Boltzmann equation with boundary conditions which is the radiosity equation in case of homogeneous media by piece.

Secondly, we introduce geometrical tools used in CSG (*Constructive Solid Geometry*) and BRep (*Boundary Representation*) hybrid modeller. Basic algorithms looking for intersections between half-lines and geometrical objects are described.

Then, we present a survey of acceleration methods in radiometric computation by spatial localization. Taking into account the constraints relevant to the industry, such an acceleration method is adapted to our context and then developed in an existing software environment. Numerical tests show the efficiency of the new libraries.

Finally, we study the time and memory complexities of spatial localization methods. This theoretical study uses Minkowski's sums of geometrical sets. An important result is a strategical method minimizing time complexity to choose the localization parameters.

Keywords : hierarchical space subdivision, complexity, Boltzmann equation, radiometry, ray tracing, geometric modeling, CAGD, CSG, BRep, Minkowski's sums

Table des matières

Introduction	1
Liste des principaux symboles	5
1 Éléments de physique	9
1.1 Modèle physique du transport	9
1.1.1 Équation du transport	11
1.1.2 Milieu homogène	15
1.1.3 Prise en compte d'une surface	18
1.1.4 Cas particuliers	19
1.2 Grandeurs radiométriques	24
1.2.1 Luminance	24
1.2.2 Émittance et éclairement	24
1.2.3 Intensité optique	25
1.2.4 Flux	25
1.2.5 Noyau de dispersion surfacique	26
1.3 Conclusion	29
2 Représentation des objets et algorithmes de base	31
2.1 Complexités	31
2.1.1 Complexité en mémoire	32
2.1.2 Complexité en temps	32
2.2 Modèle géométrique CSG	36
2.2.1 Description	36
2.2.2 Utilisation du modèle pour l'optique	40
2.2.3 Test d'intériorité	42
2.2.4 Intersection entre un rayon et un objet	43
2.2.5 Équivalence avec un arbre binaire	47
2.2.6 Complexité en mémoire	50
2.2.7 Complexité en temps	52
2.3 Modèle géométrique BRep	72
2.3.1 Description de la topologie	73
2.3.2 Description de la géométrie	74
2.3.3 Utilisation du modèle pour l'optique	75
2.3.4 Intersection entre un rayon et un objet	76
2.3.5 Complexité en mémoire	76
2.3.6 Complexité en temps	78
2.4 Intersecteurs rayon-surface	80
2.4.1 Surfaces implicites	80
2.4.2 Surfaces paramétriques	83

2.5	Conclusion	86
3	Localisation spatiale pour le calcul d'intersection	89
3.1	Accélération des calculs d'intersection	89
3.1.1	Où gagner du temps?	89
3.1.2	Utilisation d'objets englobants	91
3.1.3	Localisation spatiale	95
3.1.4	Choix d'une méthode	108
3.2	Localisation par subdivision spatiale	109
3.2.1	Subdivision uniforme	110
3.2.2	Hierarchisation	117
3.3	Résultats expérimentaux	121
3.3.1	Prototype	121
3.3.2	Environnement logiciel	122
3.3.3	Analyse des résultats	133
3.4	Conclusion	137
4	Évaluation des complexités dans une subdivision uniforme	139
4.1	Choix classiques des paramètres de subdivision	139
4.2	Nombre de régions traversées par un objet	141
4.2.1	Caractérisation	141
4.2.2	Objet linéique	144
4.2.3	Surface optique	149
4.2.4	Boîte englobante	161
4.3	Complexité en mémoire	164
4.3.1	Caractérisation	164
4.3.2	Estimation	165
4.4	Complexité en temps	169
4.4.1	Caractérisation	169
4.4.2	Ensemble d'objets	171
4.4.3	Ensemble de rayons	172
4.4.4	Longueur d'un rayon	173
4.4.5	Nombre de régions traversées par un rayon	177
4.4.6	Nombre d'objets rencontrés par un rayon	180
4.4.7	Estimation	181
4.5	Une nouvelle méthode de choix des paramètres de subdivision	184
4.6	Mise en place d'une hiérarchie de subdivisions uniformes	186
4.7	Conclusion	187
	Perspectives	189
	Bibliographie	191

Introduction

Cette thèse s'inscrit dans le contexte d'un projet proposé conjointement par la société Optis, spécialisée dans les études d'optique, et par l'équipe Modélisation Géométrique et Approximation du Laboratoire de Modélisation et Calcul, dans le cadre d'une convention Cifre subventionnée par l'Association Nationale de la Recherche Technique. Ce projet s'inscrit dans le cadre simulation et développement logiciel de la société Optis.

Un système optique est l'ensemble des organes d'un instrument qui vont de la production d'un rayonnement à sa détection. Aujourd'hui, ces systèmes jouent un rôle de plus en plus important dans l'industrie ainsi que dans divers domaines scientifiques. La conception des systèmes d'éclairage de bâtiments ou d'installations publiques, les tableaux de bord ou les phares de voiture, les façades des téléphones mobiles ou d'autres appareils du genre ou encore des machines lasers sophistiquées sont quelques exemples de tels systèmes.

La société Optis est spécialisée dans la simulation des phénomènes physiques de l'optique. C'est pourquoi elle s'intéresse en particulier à la radiométrie, discipline qui concerne la caractérisation théorique et expérimentale des rayonnements dans les systèmes optiques. L'image de synthèse, dont l'objet est la construction d'images réalistes au sens de la perception de l'oeil humain, est un domaine qui se rapproche des préoccupations des opticiens qui sont de caractériser ou évaluer au plus juste les grandeurs physiques mises en jeu. En effet, la simulation des phénomènes physiques de la radiométrie est, depuis une quinzaine d'années, également centre d'intérêt du domaine de la synthèse d'images : les performances des moyens de calcul autorisent l'utilisation de modèles physiques sophistiqués ce qui procure un réalisme accru des images ainsi produites. C'est aussi l'un des objectifs de la collaboration que de rapprocher les méthodes de la modélisation physique et de la synthèse d'image au sein de la société Optis.

La conception d'un système optique nécessite une connaissance précise des propriétés optiques des matériaux considérés et du modèle de propagation de la lumière mais aussi de la géométrie des différents éléments du système. Cette phase est coûteuse, en grande partie à cause d'une mise au point très longue qui nécessite la construction de prototypes. La simulation numérique est devenue un outil indispensable pour pallier ce problème et permettre la conception de systèmes optiques dans un temps raisonnable et à moindre coût. Cependant, les différentes méthodes numériques envisagées (en particulier Monte Carlo et radiosité) nécessitent toujours une définition précise de la géométrie des objets et fait intervenir de très nombreux calculs d'intersection entre des rayons et des objets. Ainsi, les simulations sont soumises à des temps de calcul qui ne sont plus satisfaisants à la vue des contraintes industrielles actuelles.

Il est donc indispensable pour les opticiens de développer des outils visant à accélérer les calculs dans les simulations. Ce développement devant se faire dans un contexte industriel, il est soumis à un cahier des charges strict. Les grandes lignes de ce projet s'articulent autour des axes suivants.

- L'étude des éléments de la physique sous-jacents est une nécessité pour la compréhension des phénomènes à considérer.
- L'état de l'art des modèles géométriques utilisés ainsi que des structures de données associées et des algorithmes intervenant dans les simulations permet de définir précisément le contexte

technique du projet.

- Une étude des méthodes d'accélération géométrique existantes est un élément indispensable pour nous guider dans le choix d'un outil à intégrer dans un environnement existant.

Pour chaque axe considéré, une présentation accessible doit permettre à l'entreprise une bonne compréhension et un suivi des domaines abordés. Au cours du déroulement de ce travail, les objectifs ont du évoluer pour plusieurs raisons :

- l'évolution de la modélisation géométrique utilisée par l'entreprise, d'un modèle CSG (*Constructive Solid Geometry*) vers un modèle BRep (*Boundary Representation*), due par l'évolution de la culture d'entreprise et de la demande du marché,
- l'évolution de la composition des équipes de travail au sein de l'entreprise.

Ainsi, pour prendre en compte au mieux ces évolutions, le travail s'est orienté vers l'accélération des calculs géométriques par localisation spatiale hiérarchique des objets géométriques.

Le chapitre 1 donne les éléments de base nécessaires à la définition du modèle physique du transport dans un cadre mono-énergétique. La propagation de la lumière est décrite par l'équation de Boltzmann accompagnée de conditions aux limites. Cette modélisation constitue une généralisation de l'équation de radiosité en image de synthèse, dont la première formulation dans ce contexte est due à Kajjiya [52]. Pour le projet qui nous intéresse, les milieux considérés sont homogènes par morceaux. Ainsi, on se place dans le contexte de la radiométrie, où la lumière est décrite par la luminance et les interactions avec la matière sont caractérisées par des fonctions appelées *Bidirectional Surface-Scattering Distribution Function*.

Le chapitre 2 a pour but de préciser le contexte technique du projet. La première phase d'une simulation nécessite la définition d'un système optique. Conformément à l'évolution des outils de l'entreprise, des modélisateurs géométriques, de type CSG dans un premier temps, puis mixte CSG-BRep par la suite, sont utilisés pour définir la géométrie des objets. Les structures de données associées sont enrichies pour la définition des propriétés optiques. La seconde phase consiste alors en la résolution des équations données au chapitre 1, ce qui, pour la modélisation considérée, se traduit principalement par des calculs d'intersection entre des rayons et des objets. Pour chaque modèle géométrique considéré, une étude des complexités en mémoire et en temps est effectuée dans le but de comprendre et d'analyser les faiblesses de l'environnement existant et d'en tirer des conclusions pour la mise en place d'un outil d'accélération des calculs.

Le chapitre 3 traite des méthodes d'accélération classiques pour le calcul d'intersection entre des rayons et des objets puis propose un choix de méthode en respect du cahier des charges imposé par l'entreprise (paragraphe 3.1.4). Une méthode de localisation par subdivision spatiale hiérarchique s'est avérée être le choix le plus judicieux dans ce contexte. Le développement de plusieurs bibliothèques d'accélération, suivant ainsi l'évolution des outils de la société, a permis d'obtenir des performances très intéressantes, atteignant ainsi un facteur d'accélération des temps de calcul de l'ordre de 100 dans un environnement prototype de développement et de l'ordre de 25 dans l'environnement logiciel Speos, limité par le nombre d'objets que l'on peut considérer.

Le chapitre 4, plus académique dans ses objectifs, s'intéresse à l'évaluation a priori des complexités des structures de données et des algorithmes liés à la localisation spatiale par subdivision uniforme présentés dans le chapitre 3. En effet, la méthode de localisation envisagée présente de bonnes performances mais reste soumise à un ensemble de paramètres difficiles à définir par les méthodes classiques et qui changent beaucoup suivant le type de système optique considéré. Des formulations précises des complexités en temps et en mémoire permettent de définir une stratégie de localisation visant à diminuer au mieux les temps de simulation, tout en contrôlant l'occupation mémoire.

Cleary et Wyvill [18] ont donné des expressions des complexités en fonction de quelques caractéristiques géométriques des objets (segment, polygone plan) pour des subdivisions uniformes cubiques. Dans ce chapitre une généralisation de ces résultats est proposée. Ainsi, pour prendre en compte les caractéristiques des objets considérés (géométriques : objet linéique, surfacique ou volumique ; physique : opacité) ainsi que des subdivisions uniformes non nécessairement cubiques, des formulations des complexités sont proposées. En particulier, des estimations de certains paramètres essentiels de ces formulations sont données :

- le nombre de régions traversées par un rayon ou par un objet,
- la longueur d'un rayon,
- le nombre d'objets rencontrés par un rayon.

L'exploitation de ces complexités pour le choix des paramètres de subdivision se ramène alors à un problème de minimisation à trois variables (les nombres de subdivisions dans chaque direction).

Ce projet a constitué une passerelle entre une entreprise en quête d'outils novateurs développés dans le milieu de la recherche et un laboratoire intéressé par le transfert de technologie vers l'industrie. Un des objectifs prioritaires d'une petite société en pleine croissance est d'obtenir très rapidement des bibliothèques de code, ce qui n'est pas toujours compatible avec une activité de recherche appliquée, et, de plus, implique de travailler dans des environnements en constante évolution, souvent instables. Cependant, malgré ces contraintes, le projet a pu être mené à bien, l'entreprise bénéficiant d'outils très satisfaisants. D'autre part, une recherche théorique originale a permis de mieux comprendre des méthodes très largement utilisées dans d'autres domaines, en particulier en image de synthèse.

Les différentes parties de ce travail ne constituent pas une fin en soi, et laissent même envisager d'autres travaux. Ces perspectives sont présentées en fin de document.

Liste des principaux symboles

Unités fondamentales

$[s]$	seconde
$[m]$	mètre
$[sr]$	stéradian
$[J]$	Joule
$[W]$	Watt

Photon

x	position dans un domaine $X \in \mathbb{R}^3$	-
v	vitesse	$[ms^{-1}]$
w	direction de la vitesse	-
λ	longueur d'onde	$[m]$
E_λ	énergie	$[J]$
c	vitesse dans le vide	299 792 458 $[ms^{-1}]$
h	constante de Planck	$6.625 \cdot 10^{-34} [Js]$

Équation du transport

n_p	densité angulaire de photons	$[m^{-3}sr^{-1}]$
ϕ	densité angulaire de flux	$[m^{-2}sr^{-1}s^{-1}]$
q	densité angulaire de photons émis par unité de temps	$[m^{-3}sr^{-1}s^{-1}]$
q_b	densité angulaire surfacique de photons émis par unité de temps	$[m^{-2}sr^{-1}s^{-1}]$
k	noyau de dispersion volumique	$[m^{-1}sr^{-1}]$
k_b	noyau de dispersion surfacique	$[sr^{-1}]$
σ_a	coefficient d'absorption	$[m^{-1}]$
σ_s	coefficient de dispersion	$[m^{-1}]$
σ	coefficient d'interaction totale ($\sigma = \sigma_a + \sigma_s$)	$[m^{-1}]$
g	gain	$[m^{-3}sr^{-1}s^{-1}]$
g_b	gain surfacique	$[m^{-2}sr^{-1}s^{-1}]$
τ	chemin d'absorption	-

Grandeurs radiométriques

Φ	flux	$[W]$
I	intensité optique	$[Wsr^{-1}]$
L	luminance	$[Wm^{-2}s^{-1}]$
ε	luminance émise	$[Wm^{-2}s^{-1}]$
ε_b	luminance surfacique émise	$[Wm^{-2}s^{-1}]$
η	indice de réfraction optique	-
M	émittance	$[Wm^{-2}]$
E	éclairage	$[Wm^{-2}]$
ρ	BSSDF (<i>Bidirectional Surface-Scattering Distribution Function</i>)	$[sr^{-1}]$
ρ_r	BRDF (<i>Bidirectional Reflectance Distribution Function</i>)	$[sr^{-1}]$
ρ_t	BTDF (<i>Bidirectional Transmission Distribution Function</i>)	$[sr^{-1}]$

Rayon

X	domaine de \mathbb{R}^3
x	origine dans X
\mathcal{S}^2	sphère unité sur \mathbb{R}^3
w	direction dans \mathcal{S}^2
$R_{x,w}$	rayon d'origine x et de direction w
ν_Σ	fonction distance aux surfaces Σ
ξ_Σ	fonction de lancer de rayons par rapport aux surfaces Σ
\mathcal{R}	ensemble de couples (x, w) (dans $X \times \mathcal{S}^2$)
$\mu(x)$	mesure de Lebesgue sur \mathbb{R}^3 d'un voisinage du point x
$\sigma(w)$	mesure de Lebesgue sur \mathcal{S}^2 d'un voisinage de la direction w
$\text{mes}(\mathcal{R})$	mesure de l'ensemble \mathcal{R}

Objet géométrique

O	objet géométrique
$\Sigma(O)$	surface optique associée à l'objet O
$\Upsilon(O)$	objet de localisation englobant associé à l'objet O
$B(O)$	boite englobante de localisation associée à l'objet O
Σ_o	ensemble des surfaces optiques issues des objets opaques
Σ_{no}	ensemble des surfaces optiques issues des objets non opaques
Σ	ensemble des surfaces optiques ($\Sigma = \Sigma_o \cup \Sigma_{no}$)
$\lambda_O(x, w)$	nombre d'intersection entre un rayon $R_{x,w}$ et un objet O

Objet géométrique CSG

\mathcal{O}_p	opérateur pour un objet CSG
G	opérateur géométrique
E	opérateur ensembliste
C	opérateur constructif
ξ	proportion d'opérateurs géométriques
κ	nombre de surfaces virtuelles

Subdivision uniforme

R	parallélépipède à subdiviser
∂R	bord de R
Δ_i	longueur d'un parallélépipède dans la direction i
\mathcal{S}	ensemble des régions d'une subdivision uniforme
R_k	région d'une subdivision uniforme
n	nombre de régions dans une subdivision
n_i	nombre de subdivisions dans la direction i
δ_i	longueur d'une région dans la direction i

Complexités

T	complexité en temps
M	complexité en mémoire
m_O	nombre de régions traversées par un objet O
$\bar{\cdot}$	moyenne par rapport à \mathcal{R}
$\langle \cdot \rangle$	moyenne par rapport à l'ensemble des directions de \mathcal{S}^2

Chapitre 1

Éléments de physique

Comment définir la lumière ? Nous pouvons considérer que la lumière est un phénomène de transport d'information. Il existe deux manières de transporter de l'information entre deux points, soit par un échange d'énergie, soit par un échange de masse. C'est autour de ces deux manières de concevoir le transport d'information que se sont développées les différentes théories tentant d'expliquer la nature de la lumière.

Dans la théorie ondulatoire de l'électromagnétique, les équations de Maxwell donnent une description énergétique de la nature de la lumière. Des hypothèses simplificatrices faites sur ces équations permettent de passer successivement à l'équation des ondes, à l'équation eikonale (ou iconale), puis, lorsque la longueur d'onde considérée est très petite, à l'équation du rayon lumineux. C'est à partir de cette dernière qu'est construite l'optique géométrique (Bruhat [12], May [65], Bouche et Molinet [8]). La description particulière de la nature de la lumière conduit à l'équation de Boltzmann de la théorie du transport. Les bases physiques de ce modèle se trouvent dans la théorie du transfert thermique (Siegel et Howell [77]). La neutronique est un domaine qui utilise ce modèle physique et dont la formalisation mathématique des méthodes de résolution a été exposée par Dautray et Lions [26] [25]. Depuis quelques années, les chercheurs du domaine de l'image de synthèse portent une attention particulière à ce modèle physique Arvo et Kirk [4], Arvo [2] [3], Cohen et Wallace [20], Glassner [44], Sillion et Puech [78] pour ne citer que ces auteurs.

Cette dernière description de la lumière est le modèle que nous allons considérer tout au long de ce travail. Il est satisfaisant pour les phénomènes qui doivent être pris en compte dans le contexte industriel. Le présent chapitre est inspiré des différents ouvrages et articles cités en référence. Il a pour objet de rappeler comment l'équation du transport est obtenue.

1.1 Modèle physique du transport

L'équation du transport décrit l'évolution d'une population de particules dans un domaine $X \subset \mathbb{R}^3$ occupé par un milieu en interaction avec les particules. Le système est supposé fermé, ce qui signifie qu'aucune influence externe au domaine X n'est considérée. En d'autres termes, les conditions de bord sur ∂X sont :

- une particule sortant de X est perdue,
- aucune particule n'entre dans X .

La lumière peut-être considérée comme étant constituée de particules, appelées photons. Un photon est décrit par :

- sa position : $x \in X$,
- la direction de sa vitesse v : $w \in \mathcal{S}^2$ (sphère unité dans \mathbb{R}^3),
- sa longueur d'onde : $\lambda > 0$.

Les travaux de Planck et d'Einstein ont conduit à admettre que l'énergie d'un photon est quantifiée :

$$E_\lambda = \frac{h\nu}{\lambda} \quad [J] ,$$

où h est la constante de Planck.

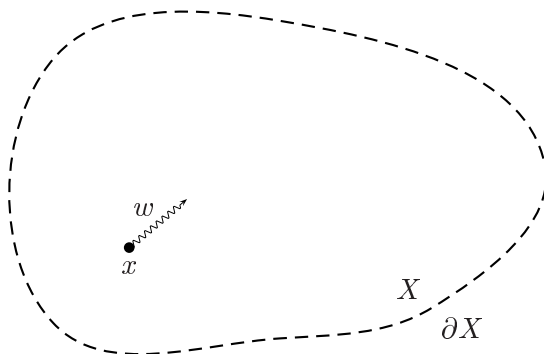


Fig. 1.1 – Notations.

L'analyse de la propagation de la lumière consiste à étudier les déplacements et les interactions avec le milieu d'un ensemble de photons. Pour caractériser cette propagation, il est nécessaire de spécifier les degrés de liberté d'un ensemble de photons à un instant donné. La densité angulaire de photons, notée $n_p(x, w)$, est une fonction à valeurs dans \mathbb{R} , d'unité $[m^{-3}sr^{-1}]$, définie telle que :

$$n_p(x, w) d\mu(x) d\sigma(w)$$

soit le nombre de photons dans le volume différentiel $d\mu(x)$ autour du point x se déplaçant dans une direction contenue dans l'angle solide différentiel $d\sigma(w)$ autour de la direction w .

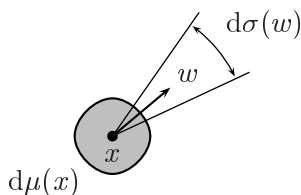


Fig. 1.2 – Densité angulaire de photons.

Il est souvent plus intéressant de caractériser un flux de photons qui traverse une surface réelle ou imaginaire. La densité angulaire de flux, notée $\phi(x, w)$ est une fonction à valeurs dans \mathbb{R} définie par :

$$\phi(x, w) = v n_p(x, w) \quad [m^{-2}sr^{-1}s^{-1}] .$$

Ainsi, le nombre de photons traversant une surface différentielle $d\mu(x)$ autour du point x se déplaçant dans une direction contenue dans l'angle solide différentiel $d\sigma(w)$ autour de la direction w , par unité de temps, est :

$$\phi(x, w) \cos \theta d\mu(x) d\sigma(w) ,$$

où θ est l'angle entre le vecteur normal à la surface et la direction w .

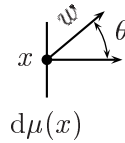


Fig. 1.3 – Nombre de photons à travers une surface.

Les phénomènes que nous allons considérer dans la suite permettent de rajouter deux hypothèses au modèle physique. Tout d'abord, les milieux considérés sont isotropes, c'est-à-dire que les propriétés de ces milieux ne dépendent pas de la direction. Ensuite, la lumière considérée est non cohérente, c'est-à-dire que les photons qui la composent n'interagissent pas entre eux. L'étude est donc menée dans un cadre mono-énergétique, ce qui signifie que les densités angulaires de flux pourront être étudiées de manière indépendante pour chaque longueur d'onde considérée. Pour obtenir la contribution d'un ensemble de longueurs d'onde, il suffit d'écrire que la densité angulaire de flux résultante est la somme des densités angulaires de flux de chaque longueur d'onde :

$$\phi(x, w) = \int_0^{+\infty} \phi_\lambda(x, w) d\lambda,$$

où $\phi_\lambda(x, w)$ est la densité angulaire de flux au point (x, w) pour la longueur d'onde λ . Dans la suite, sauf exception, une seule longueur d'onde sera prise en compte. Ainsi, la longueur d'onde λ sera omise.

Sous l'hypothèse d'une étude mono-énergétique, un photon possède donc 5 degrés de liberté (3 pour la position x dans \mathbb{R}^3 et 2 pour la direction w dans \mathcal{S}^2). Il évolue dans un espace inclus dans $\mathbb{R}^3 \times \mathcal{S}^2$, appelé espace des phases. Les caractéristiques de cet espace sont précisées dans le paragraphe 2.1.2. En particulier, la définition d'une mesure sur l'espace des phases est induite par les mesures de Lebesgue sur \mathbb{R}^3 et \mathcal{S}^2 , notées respectivement μ et σ . Ainsi, la forme différentielle $d\mu$ correspond à une surface ou un volume élémentaire. De même, la forme différentielle $d\sigma$ correspond à un angle solide élémentaire, c'est-à-dire l'aire de la portion de sphère unité interceptée par un angle solide élémentaire.

1.1.1 Équation du transport

Soit $Y \subset X$ un volume et $\Omega \subset \mathcal{S}^2$ un angle solide. L'équation du transport s'établit en déterminant les variations du nombre de photons dans $Y \times \Omega$ par unité de temps. Pour ce faire, procédons en deux étapes. La première étape consiste à identifier les phénomènes physiques mis en jeu. Ils peuvent être classés en trois catégories :

- photons émis dans $Y \times \Omega$,
- photons traversant $Y \times \Omega$,
- collisions entre photons et matière dans $Y \times \Omega$.

La seconde étape consiste à regrouper les expressions obtenues pour ces phénomènes physiques, dans le but d'écrire l'état d'équilibre qui conduit à l'équation du transport.

Photons émis

Le premier phénomène est l'émission de photons. La densité angulaire de photons émis, d'unité $[m^{-3}sr^{-1}s^{-1}]$, notée $q(x, w)$, est une fonction à valeurs dans \mathbb{R} définie telle que :

$$q(x, w) d\mu(x) d\sigma(w)$$

soit le nombre de photons émis dans le volume différentiel $d\mu(x)$ autour du point x se déplaçant dans une direction contenue dans l'angle solide différentiel $d\sigma(w)$ autour de la direction w , par unité de temps. Ainsi, le nombre de photons émis par unité de temps dans $Y \times \Omega$ est (figure 1.4) :

$$E = \int_{\Omega} \int_Y q(x, w) d\mu(x) d\sigma(w) \quad [s^{-1}] .$$

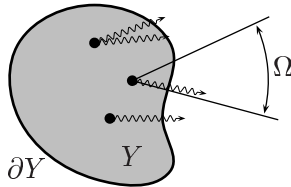


Fig. 1.4 – Photons émis par unité de temps.

Photons traversants

Le nombre de photons traversant $Y \times \Omega$ (*streaming*), est le nombre de photons qui traversent la surface ∂Y , frontière du volume Y , dans un angle solide Ω , par unité de temps (figure 1.5) :

$$S = \int_{\Omega} \int_{\partial Y} \langle \phi(x, w) w, u(x) \rangle d\mu(x) d\sigma(w) \quad [s^{-1}] ,$$

où $u(x)$ est le vecteur normal unitaire sortant de ∂Y au point x et la notation $\langle \cdot, \cdot \rangle$ représente le produit scalaire dans \mathbb{R}^3 . Les photons qui entrent dans ∂Y contribuent à la diminution de S , et ceux qui sortent contribuent à l'augmentation de S .

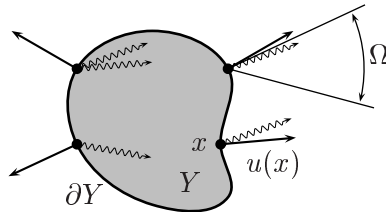


Fig. 1.5 – Photons traversants par unité de temps.

Collisions

Les collisions entre les photons et la matière peuvent être séparés en deux catégories : absorption et dispersion. Les hypothèses du modèle du transport des photons impliquent que les collisions ne se produisent qu'entre des photons et la matière, les photons n'interagissent pas entre eux, qu'elles sont indépendantes de la densité de flux angulaire, et qu'elles sont proportionnelles à la distance parcourue par le photon.

Le coefficient d'absorption, d'unité $[m^{-1}]$, notée $\sigma_a(x)$, est une fonction à valeurs dans \mathbb{R} définie telle que :

$$\sigma_a(x) n_p(x, w) d\mu(x) d\sigma(w)$$

soit le nombre de photons absorbés dans le volume différentiel $d\mu(x)$ autour du point x se déplaçant dans une direction contenue dans l'angle solide différentiel $d\sigma(w)$ autour de la direction w , par unité de longueur traversée. Ainsi, le nombre de photons absorbés par unité de temps dans $Y \times \Omega$ est (figure 1.6) :

$$C_{\text{abs}} = \int_{\Omega} \int_Y \sigma_a(x) \phi(x, w) d\mu(x) d\sigma(w) \quad [s^{-1}] .$$

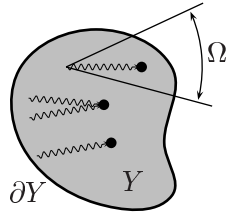


Fig. 1.6 – Photons absorbés par unité de temps.

La dispersion est un phénomène beaucoup plus complexe. Elle est décrite par un noyau de dispersion volumique, d'unité $[m^{-1}sr^{-1}]$, noté $k(x, \langle w, w' \rangle)$, à valeurs dans $[0, 1]$, tel que :

$$k(x, \langle w, w' \rangle) n_p(x, w) d\sigma(w') d\mu(x) d\sigma(w)$$

soit le nombre de photons se trouvant dans le volume différentiel $d\mu(x)$ autour du point x se déplaçant dans une direction contenue dans l'angle solide différentiel $d\sigma(w)$ autour de la direction w qui sont déviés dans une direction contenue dans l'angle solide différentiel $d\sigma(w')$ autour de la direction w' , par unité de longueur traversée.

Remarque

De manière générale, le noyau de dispersion volumique est noté $k(x, w \rightarrow w')$. Mais, dans le cas particulier d'un matériau isotrope, les directions w et w' ne sont pas significatives. Ce qui entre en compte c'est l'angle entre ces deux vecteurs. Il est caractérisé par le produit scalaire $\langle w, w' \rangle$. C'est pourquoi :

$$k(x, w \rightarrow w') = k(x, w' \rightarrow w) = k(x, \langle w, w' \rangle) .$$

□

Le premier phénomène de dispersion (*in-scatter*) comptabilise les photons de Y qui sont déviés dans une direction appartenant à Ω (figure 1.7). Le nombre de photons par unité de temps dispersés de la sorte est :

$$C_{\text{in}} = \int_{\Omega} \int_Y \int_{S^2} k(x, \langle w', w \rangle) \phi(x, w') d\sigma(w') d\mu(x) d\sigma(w) \quad [s^{-1}] .$$

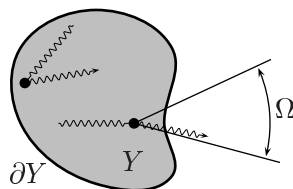


Fig. 1.7 – Photons dispersés dans Ω par unité de temps.

Le second phénomène de dispersion (*out-scatter*) est en quelque sorte l'inverse du précédent. Il comptabilise le nombre de photons dont la direction est dans Ω et qui sont déviés dans une autre direction (figure 1.8). Le nombre de photons par unité de temps dispersés de la sorte est :

$$C_{\text{out}} = \int_{\Omega} \int_Y \int_{S^2} k(x, \langle w, w' \rangle) \phi(x, w) d\sigma(w') d\mu(x) d\sigma(w) \quad [s^{-1}] .$$

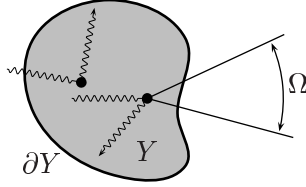


Fig. 1.8 – Photons dispersés en dehors de Ω par unité de temps.

Équation du transport

Les cinq expressions obtenues pour les phénomènes physiques mis en jeu décrivent une variation du nombre de photons dans $Y \times \Omega$ par unité de temps. Écrire l'état d'équilibre, c'est comptabiliser les photons entrant dans $Y \times \Omega$ et les photons sortant de $Y \times \Omega$ par unité de temps :

$$\begin{aligned} & \int_{\Omega} \int_Y q(x, w) d\mu(x) d\sigma(w) \\ & \quad + \\ & \int_{\Omega} \int_Y \int_{S^2} k(x, \langle w', w \rangle) \phi(x, w') d\sigma(w') d\mu(x) d\sigma(w) \\ & \quad = \\ & \int_{\Omega} \int_{\partial Y} \langle \phi(x, w) w, u(x) \rangle d\mu(x) d\sigma(w) \\ & \quad + \\ & \int_{\Omega} \int_Y \sigma_a(x) \phi(x, w) d\mu(x) d\sigma(w) \\ & \quad + \\ & \int_{\Omega} \int_Y \int_{S^2} k(x, \langle w, w' \rangle) \phi(x, w) d\sigma(w') d\mu(x) d\sigma(w) \end{aligned}$$

ce que l'on peut noter plus succinctement :

$$\boxed{\underbrace{E + C_{\text{in}}}_{\text{entrants}} = \underbrace{S + C_{\text{abs}} + C_{\text{out}}}_{\text{sortants}}} . \quad (1.1)$$

L'équation 1.1 traduit un état d'équilibre sur un domaine $Y \times \Omega$, où Y et Ω sont arbitraires. Pour écrire une équation ponctuelle, il est nécessaire de transformer le terme S . Le théorème de Gauss permet de passer d'une intégration sur ∂Y à une intégration sur Y :

$$\begin{aligned} S &= \int_{\Omega} \int_{\partial Y} \langle \phi(x, w) w, u(x) \rangle d\mu(x) d\sigma(w) \\ &= \int_{\Omega} \int_Y \langle w, \nabla_x \phi(x, w) \rangle d\mu(x) d\sigma(w) . \end{aligned}$$

Ainsi, en chaque point $(x, w) \in Y \times \Omega$, la densité de flux angulaire $\phi(x, w)$ vérifie l'équation suivante :

$$q(x, w) + \int_{\mathcal{S}^2} k(x, \langle w', w \rangle) \phi(x, w') d\sigma(w') = \langle w, \nabla_x \phi(x, w) \rangle + \sigma_a(x) \phi(x, w) + \phi(x, w) \int_{\mathcal{S}^2} k(x, \langle w, w' \rangle) d\sigma(w'). \quad (1.2)$$

Pour simplifier l'écriture de cette équation introduisons deux coefficients caractéristiques du milieu. Le coefficient de dispersion, d'unité $[m^{-1}]$, notée $\sigma_s(x)$, est une fonction à valeurs dans \mathbb{R} définie telle que :

$$\sigma_s(x) n_p(x, w) d\mu(x) d\sigma(w)$$

soit le nombre de photons dispersés dans le volume différentiel $d\mu(x)$ autour du point x se déplaçant dans une direction contenue dans l'angle solide différentiel $d\sigma(w)$ autour de la direction w , par unité de longueur traversée. Elle est définie par :

$$\sigma_s(x) = \int_{\mathcal{S}^2} k(x, \langle w_0, w' \rangle) d\sigma(w'),$$

où w_0 est une direction quelconque, puisque le milieu est isotrope. Le coefficient d'interaction totale, d'unité $[m^{-1}]$, notée $\sigma(x)$, est une fonction à valeurs dans \mathbb{R} définie par :

$$\sigma(x) = \sigma_a(x) + \sigma_s(x).$$

Ainsi, l'équation du transport 1.2, appelée équation intégral-différentielle de Boltzmann, s'écrit de la manière suivante :

$$\boxed{q(x, w) + \int_{\mathcal{S}^2} k(x, \langle w', w \rangle) \phi(x, w') d\sigma(w') = \langle w, \nabla_x \phi(x, w) \rangle + \sigma(x) \phi(x, w)} \quad (1.3)$$

Le passage de 1.1 à 1.3 n'a pas de réel sens physique. D'un point de vue mathématique, l'égalité dans l'équation 1.3 doit être comprise au sens faible ou des distributions.

Remarque

Le noyau de dispersion volumique peut s'exprimer à l'aide de la fonction de phase, d'unité $[sr^{-1}]$, notée $P(x, \alpha)$, par la relation suivante :

$$k(x, \alpha) = \frac{P(x, \alpha) \sigma_s(x)}{4\pi},$$

où $\alpha \in [0, 1]$.

□

1.1.2 Milieu homogène

Indice de réfraction optique

On appelle indice de réfraction optique d'un milieu la quantité, notée η , qui caractérise la vitesse de propagation d'un photon dans ce milieu :

$$\eta = \frac{c}{v} = \frac{h c}{\lambda E_\lambda},$$

où c est la vitesse d'un photon dans le vide. Par définition, un milieu est dit homogène si l'indice de réfraction optique est constant en tout point de ce milieu. η est alors une caractéristique du milieu.

Plus un milieu est dense, plus son indice de réfraction optique est élevé, et, par conséquent, moins la vitesse d'un photon qui le traverse est grande.

L'indice de réfraction optique peut être vu comme une fonction de la position, et on notera $\eta(x)$ sa valeur en un point x de X . Dans le cadre de notre travail, cette fonction peut être considérée comme constante par morceaux. Le lieu des points de discontinuité de l'indice de réfraction optique décrit alors des surfaces, qui jouent un rôle fondamental en optique. Une surface peut éventuellement délimiter un volume fermé. On note Σ l'ensemble des surfaces de X (figure 1.9).

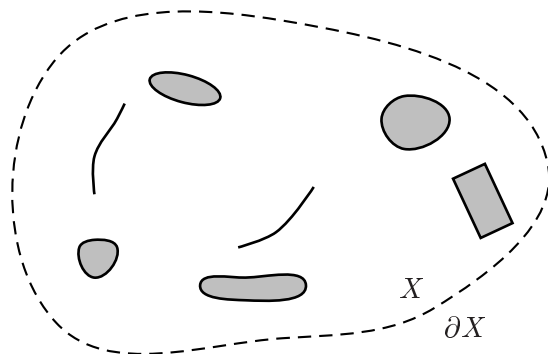


Fig. 1.9 – Ensemble des surfaces dans le domaine X (Σ est l'ensemble des points appartenant aux traits en gras).

Intégration le long d'un rayon

Soit un milieu homogène, et soit un point x dans ce milieu. L'optique géométrique, et plus précisément le théorème de Fermat¹, précise que les photons suivent le plus court chemin entre deux points. Le milieu considéré est homogène, le chemin est donc un segment de droite, appelé rayon.

L'équation de Boltzmann 1.3 est une équation différentielle linéaire du premier ordre en la variable x . Le but de ce paragraphe est de donner une expression de la densité angulaire de flux au point (x, w) en fonction de la densité angulaire de flux en un point situé sur le rayon considéré et des contributions le long de ce rayon (figure 1.10).

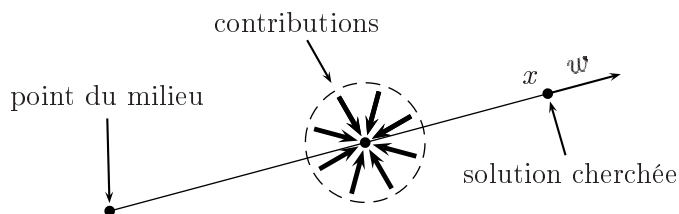


Fig. 1.10 – Contributions dues à l'équation du transport le long d'un rayon dans un milieu homogène.

Soit \hat{f} une fonction définie sur \mathbb{R} associée à une fonction f définie sur $\mathbb{R}^3 \times \mathcal{S}^2$ de la manière suivante :

$$\hat{f}(t) = f(x - tw, w) .$$

De manière analogue, soit \hat{f} une fonction définie sur \mathbb{R} associée à une fonction f définie sur \mathbb{R}^3 de la

¹Voir les ouvrages généraux, tels que ceux écrit par Bruhat [12] ou May [65].

manière suivante :

$$\hat{f}(t) = f(x - t w) .$$

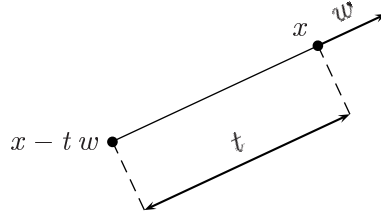


Fig. 1.11 – Définition de la variable t .

La première étape consiste à simplifier l'opérateur nabla en écrivant le produit scalaire sous la forme d'une dérivée par rapport à une variable :

$$\begin{aligned} \langle w, \nabla_x \phi(x, w) \rangle &= \left. \frac{\partial}{\partial t} \phi(x + t w, w) \right|_{t=0} \\ &= - \left. \frac{\partial}{\partial t} \phi(x - t w, w) \right|_{t=0} , \end{aligned}$$

où $t > 0$. Finalement, en utilisant la notation $\hat{\phi}$, on obtient :

$$\langle w, \nabla_x \phi(x, w) \rangle = - \frac{d}{dt} \hat{\phi}(0) . \quad (1.4)$$

Pour simplifier les écritures, soit le gain une fonction à valeurs dans \mathbb{R} , notée $g(x, w)$, définie pour tout $(x, w) \in (X \setminus \Sigma) \times \mathcal{S}^2$ par :

$$g(x, w) = q(x, w) + \int_{\mathcal{S}^2} k(x, \langle w', w \rangle) \phi(x, w') d\sigma(w') . \quad (1.5)$$

La seconde étape consiste à injecter dans l'équation de Boltzmann 1.3, écrite au point $(x - t w, w)$, l'équation 1.4, écrite au point $(x - t w, w)$, et à remplacer les fonctions ϕ , σ et g par les fonctions chapeaux correspondantes $\hat{\phi}$, $\hat{\sigma}$ et \hat{g} :

$$\frac{d}{dt} \hat{\phi}(t) - \hat{\sigma}(t) \hat{\phi}(t) = -\hat{g}(t) . \quad (1.6)$$

L'équation 1.6 est une équation différentielle linéaire du premier ordre par rapport à la variable t . Pour en donner une solution considérons un facteur d'intégration. C'est une fonction μ qui vérifie :

$$\frac{d}{dt} [\mu(t) \hat{\phi}(t)] = \mu(t) \left[\frac{d}{dt} \hat{\phi}(t) - \hat{\sigma}(t) \hat{\phi}(t) \right] .$$

La résolution de l'équation précédente donne :

$$\mu(t) = \exp \left(- \int_0^t \hat{\sigma}(u) du \right) .$$

Ainsi, en multipliant l'équation 1.6 par $\mu(t)$ on trouve la relation suivante :

$$\frac{d}{dt} [\mu(t) \hat{\phi}(t)] = -\mu(t) \hat{g}(t) .$$

En intégrant l'équation précédente entre 0 et t on obtient :

$$\mu(t) \hat{\phi}(t) - \mu(0) \hat{\phi}(0) = - \int_0^t \mu(s) \hat{g}(s) ds ,$$

puis en remarquant que $\mu(0) = 1$ on arrive à l'équation qui nous intéresse :

$$\hat{\phi}(0) = \mu(t) \hat{\phi}(t) + \int_0^t \mu(s) \hat{g}(s) ds .$$

Il ne reste plus qu'à remplacer les fonctions chapeaux par leurs homologues, on obtient ainsi l'expression recherchée de la densité angulaire de flux au point (x, w) :

$$\boxed{\phi(x, w) = \tau(x - tw, x) \phi(x - tw, w) + \int_0^t \tau(x - sw, x) g(x - sw, w) ds} , \quad (1.7)$$

où $\tau(x - tw, x)$ est le chemin d'absorption entre les points $x - tw$ et x :

$$\tau(x - tw, x) = \exp \left(- \int_0^t \sigma(x - uw) du \right) .$$

L'équation 1.7 est appelée forme intégrale de l'équation de Boltzmann. Cette équation exprime la densité angulaire de flux en un point (x, w) en fonction de ce qui se trouve en arrière de ce point. Le premier terme rend compte du transport de la densité angulaire de flux du point $(x - tw, w)$ vers le point (x, w) . Le second terme rend compte des différentes contributions le long du rayon considéré.

Si l'on recommence les calculs à partir de $x + tw$, alors l'équation résultante est similaire dans son expression, mais exprime la densité angulaire de flux d'un point (x, w) en fonction de ce qui se trouve en avant de ce point, c'est-à-dire dans le sens inverse par rapport au cas précédent. Cela revient aussi à exprimer $\phi(x - tw, w)$ en fonction de $\phi(x, w)$ dans l'équation 1.7 et à translater la variable d'espace de tw . L'équation ainsi obtenue est la suivante :

$$\phi(x, w) = \tau^{-1}(x, x + tw) \phi(x + tw, w) + \int_0^t \tau^{-1}(x, x + sw) g(x + sw, w) ds . \quad (1.8)$$

1.1.3 Prise en compte d'une surface

L'introduction de la notion de surface conduit à la notion d'orientation. Soit x un point de ∂X ou de Σ , et soit $u(x)$ un vecteur normal unitaire au point x . L'ensemble des directions de \mathcal{S}^2 peut être séparé en trois parties, notées $\Omega^-(x)$, $\Omega^0(x)$ et $\Omega^+(x)$, telles que (figure 1.12) :

$$\begin{cases} \Omega^-(x) &= \{w \in \mathcal{S}^2, \langle w, u(x) \rangle < 0\} \\ \Omega^0(x) &= \{w \in \mathcal{S}^2, \langle w, u(x) \rangle = 0\} \\ \Omega^+(x) &= \{w \in \mathcal{S}^2, \langle w, u(x) \rangle > 0\} \end{cases} .$$

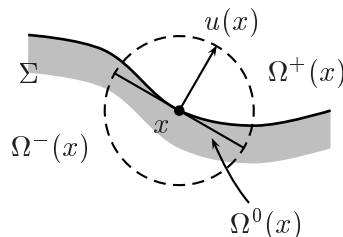


Fig. 1.12 – Décomposition de la sphère unité.

Soit x un point de Σ et soit $u(x)$ le vecteur normal unitaire au point x . Par convention, si la surface considérée au point x est fermée alors le vecteur $u(x)$ est orienté vers l'extérieur du volume délimité par cette surface. La densité angulaire de flux au niveau des surfaces est décrite par deux phénomènes. Tout d'abord, le phénomène d'émission surfacique est décrit par une fonction à valeurs dans \mathbb{R} , d'unité $[m^{-2}sr^{-1}s^{-1}]$, notée $q_b(x, w)$, telle que :

$$q_b(x, w) d\mu(x) d\sigma(w)$$

soit le nombre de photons émis dans l'élément de surface différentiel $d\mu(x)$ autour du point x se déplaçant dans une direction contenue dans l'angle solide élémentaire $d\sigma(w)$ autour de la direction w , par unité de temps.

Ensuite, le phénomène de dispersion surfacique est décrit par une fonction à valeurs dans $[0, 1]$, d'unités $[sr^{-1}]$, notée $k_b(x, w' \rightarrow w)$, appelée noyau de dispersion surfacique, telle que :

$$k_b(x, w' \rightarrow w) n_p(x, w) d\sigma(w') d\mu(x) d\sigma(w)$$

soit le nombre de photons se trouvant dans le volume différentiel $d\mu(x)$ autour du point x se déplaçant dans une direction contenue dans l'angle solide différentiel $d\sigma(w')$ autour de la direction w' qui sont déviés dans une direction contenue dans l'angle solide différentiel $d\sigma(w)$ autour de la direction w . Ce noyau décrit les phénomènes de réflexion, transmission et absorption au niveau des surfaces. Des exemples de tels noyaux sont donnés au paragraphe 1.2.5.

Ces deux phénomènes s'ajoutent, la densité angulaire de flux au niveau d'une surface s'écrit de la manière suivante :

$$\boxed{\phi(x, w) = q_b(x, w) + \int_{S^2} k_b(x, w' \rightarrow w) \phi(x, w') d\sigma(w')} \quad . \quad (1.9)$$

1.1.4 Cas particuliers

La résolution de l'équation de Boltzmann dans le cas général est un problème très difficile. C'est pourquoi on s'intéresse à des cas qui correspondent à des configurations particulières donnant lieu à une formulation simple de la modélisation du transport. Physiquement, ces cas ne sont pas réels, mais ils constituent une approche de la réalité satisfaisante pour un grand nombre d'expérimentations.

Milieu vide

Soit une surface délimitant un volume fermé Y dans le domaine X . Supposons que le milieu Y soit vide, c'est-à-dire ne contenant aucun élément susceptible d'interagir avec des photons. Ainsi, les grandeurs physiques intervenant dans le modèle du transport sont toutes nulles :

$$\begin{cases} q = 0 \\ k = 0 \\ \sigma = 0 \end{cases} \quad .$$

Les équations de Boltzmann 1.7 et 1.8 se résument donc à l'équation suivante :

$$\phi(x, w) = \phi(x - tw, w) \quad ,$$

pour tout $t \in \mathbb{R}$ tel que $x - tw \in Y$. En d'autres termes, la densité angulaire de flux est constante, et déterminée par les valeurs sur ∂Y .

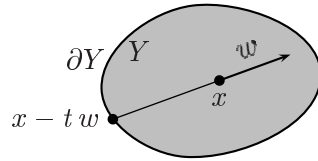


Fig. 1.13 – Transport de la densité angulaire de flux dans un milieu vide.

Milieu homogène non dispersif

Soit une surface délimitant un volume fermé Y dans le domaine X . Dire que le milieu Y est homogène et non dispersif revient à ne considérer que des propriétés d'émission et d'absorption constantes, c'est-à-dire :

$$\begin{cases} q = q_0 \\ k = 0 \\ \sigma_a = \sigma_{a,0} \end{cases} .$$

Cette hypothèse implique que le coefficient d'interaction totale σ est égal au coefficient d'absorption σ_a . L'équation de Boltzmann 1.7 s'écrit alors de la manière suivante :

$$\phi(x, w) = \tau(x - tw, x) \phi(x - tw, w) + \int_0^t \tau(x - sw, x) q(x - sw, w) ds ,$$

pour tout $t > 0$ tel que $x - tw \in \partial Y$. Puisque l'émission et l'absorption sont constantes l'équation précédente se simplifie :

$$\begin{aligned} \phi(x, w) &= \exp(-\sigma_{a,0} t) \phi(x - tw, w) + \int_0^t \exp(-\sigma_a s) q_0 ds \\ &= \exp(-\sigma_{a,0} t) \phi(x - tw, w) - \frac{q_0}{\sigma_{a,0}} (\exp(-\sigma_{a,0} t) - 1) \\ &= \left(\phi(x - tw, w) - \frac{q_0}{\sigma_{a,0}} \right) \exp(-\sigma_{a,0} t) + \frac{q_0}{\sigma_{a,0}} . \end{aligned}$$

La fonction exponentielle dans l'équation précédente décrit une atténuation de la densité angulaire de flux lors du transport dans Y .

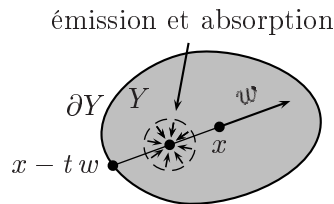


Fig. 1.14 – Transport de la densité angulaire de flux dans un milieu homogène non dispersif.

Milieu homogène par morceaux

Soient $\partial Y_1, \dots, \partial Y_N$ des surfaces délimitant éventuellement des volumes fermés (figure 1.15). L'expression de l'ensemble des surfaces est alors la suivante :

$$\Sigma = \bigcup_{i=1}^N \partial Y_i .$$

On suppose que chaque milieu correspondant à un volume fermé Y_i est homogène. De même, le milieu complémentaire dans X est homogène. Ainsi, le milieu X est homogène par morceaux (figure 1.15).

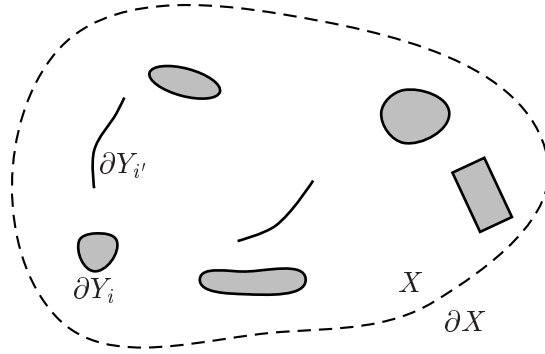


Fig. 1.15 – Surfaces dans un milieu homogène par morceaux.

Le transport de la densité angulaire de flux entre deux points (x, w) et $(x + t w, w)$, tels que le segment $[x, x + t w]$ n'ait d'intersection avec aucune des surfaces ∂Y_i , est donné par les deux cas particuliers précédents selon que le milieu est vide ou homogène non dispersif :

$$\phi(x, w) = \begin{cases} \phi(x - t w, w) & \text{si milieu vide} \\ \left(\phi(x - t w, w) - \frac{q_0}{\sigma_{a,0}} \right) \exp(-\sigma_{a,0} t) + \frac{q_0}{\sigma_{a,0}} & \text{si milieu homogène non dispersif} \end{cases} . \quad (1.10)$$

C'est pourquoi on ne s'intéresse ici qu'au transport de la densité angulaire de flux depuis une surface ∂Y_i et au transport de la densité angulaire de flux entre les différentes surfaces $\partial Y_1, \dots, \partial Y_N$. Pour ce faire, nous allons dans un premier temps exprimer la densité angulaire de flux en un point d'une surface, puis donner une relation décrivant le transport de la densité angulaire de flux entre deux points situés sur des surfaces.

Soit $x \in X \setminus \Sigma$, c'est-à-dire un point n'appartenant pas à une surface ∂Y_i . Soit $x' \in \Sigma \cup \partial X$ le premier point d'intersection entre le rayon d'origine x et de direction $-w$ et une surface ∂Y_i :

$$x' = \xi_\Sigma(x, -w) ,$$

où ξ_Σ est la fonction lancer de rayon :

$$\begin{aligned} \xi_\Sigma : X \times \mathcal{S}^2 &\longrightarrow \Sigma \cup \{x_\infty\} \\ (x, w) &\longmapsto \xi_\Sigma(x, w) = x + \nu_\Sigma(x, w) w \end{aligned} , \quad (1.11)$$

où x_∞ est un point à l'infini, qui représente le cas où il n'y a pas d'intersection, et ν_Σ est la fonction distance aux surfaces :

$$\begin{aligned} \nu_\Sigma : X \times \mathcal{S}^2 &\longrightarrow [0, +\infty] \\ (x, w) &\longmapsto \nu_\Sigma(x, w) \end{aligned} . \quad (1.12)$$

Par convention, on définit la fonction ν_Σ de la manière suivante :

$$\nu_\Sigma(x, w) = \begin{cases} t' = \inf \{t > 0, x + t w \in \Sigma \cup \partial X\} & \text{si } R_{x,w} \cap \Sigma \neq \emptyset \\ +\infty & \text{sinon} \end{cases},$$

où $R_{x,w}$ est la demi-droite orientée, ou rayon, d'origine x et de direction w . Cette définition de ν_Σ exprime le fait que les rayons qui n'ont pas d'intersection avec une surface ∂Y_i ont une intersection avec le bord du domaine considéré, c'est-à-dire ∂X , et donc n'ont aucune intersection, d'où une longueur infinie du rayon considéré.

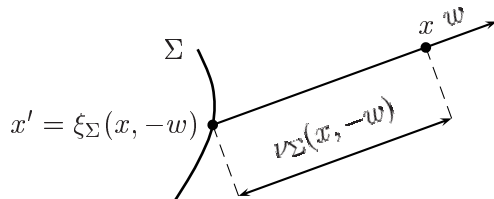


Fig. 1.16 – Définition des fonctions lancer de rayon ξ_Σ et distance aux surfaces ν_Σ .

La densité angulaire de flux au point (x, w) est donnée par la relation 1.10 en fonction de la densité angulaire de flux au point (x', w) . L'expression de la densité angulaire de flux au point (x', w) n'est autre que le gain surfacique en ce point :

$$\begin{aligned} \phi(x', w) &= g_b(x', w) \\ &= q_b(x', w) + \int_{\mathcal{S}^2} k_b(x', w' \rightarrow w) \phi(x', w') d\sigma(w') \quad . \end{aligned}$$

Soit $x \in \Sigma$, c'est-à-dire un point appartenant à une surface ∂Y_i . La densité angulaire de flux au point (x, w) est le gain surfacique en ce point :

$$\begin{aligned} \phi(x, w) &= g_b(x, w) \\ &= q_b(x, w) + \int_{\Omega(x)} k_b(x, w' \rightarrow w) \phi(\xi_\Sigma(x, -w'), w') d\sigma(w'), \quad \forall (x, w) \in \Sigma \times \mathcal{S}^2 \quad . \end{aligned}$$

Cette équation décrit un échange de photons entre les surfaces. Elle est similaire à celle donnée par Kajiyama [52]. De plus, si le noyau de dispersion surfacique est supposé indépendant des directions, alors la densité angulaire de flux ne dépend pas de la direction sur l'ensemble des surfaces :

$$\phi(x, w) = \phi(x), \quad \forall x \in \Sigma .$$

On écrit le noyau de dispersion surfacique sous la forme suivante :

$$k_b(x, \cdot \rightarrow \cdot) = \rho(x) \cos \theta' .$$

où $\rho(x)$ est la BSSDF (*Bidirectional Surface-Scattering Distribution Function*) au point x . Dans ce cas, l'équation précédente donnant la densité angulaire de flux en un point d'une surface est une version continue de l'équation de radiativité (Cohen et Wallace [20], Sillion et Puech [78]), qui consiste à rechercher la solution de l'équation de Boltzmann sur l'ensemble des surfaces :

$$\phi(x) = q_b(x) + \rho(x) \int_{\Omega(x)} \phi(\xi_\Sigma(x, -w')) \cos \theta' d\sigma(w'), \quad \forall x \in \Sigma . \quad (1.13)$$

L'équation 1.13 peut être formulée de manière différente pour faire apparaître explicitement l'échange d'énergie entre les surfaces. Pour ce faire, deux fonctions caractéristiques des surfaces vont être définies : la fonction de visibilité et le facteur géométrique. Soit x' le point de Σ défini par :

$$x' = \xi_{\Sigma}(x, -w') .$$

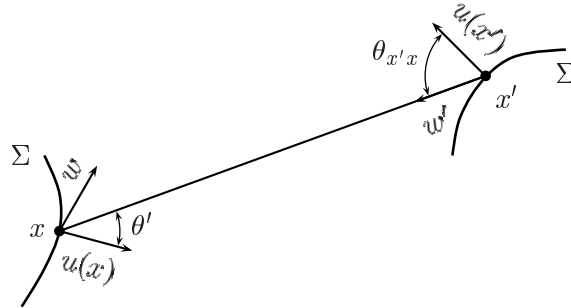


Fig. 1.17 – Échange d'énergie entre deux surfaces.

Deux points x et x' de X sont dit visibles si le segment $[x, x']$ n'a pas d'intersection avec Σ . La fonction de visibilité, notée $v_{\Sigma}(x, x')$, est une fonction à valeurs dans $\{0, 1\}$, définie par :

$$v_{\Sigma}(x, x') = \begin{cases} 0 & \text{si } [x, x'] \cap \Sigma \neq \emptyset \\ 1 & \text{sinon} \end{cases} .$$

Le facteur géométrique est une fonction à valeurs dans \mathbb{R} , notée $g_{\Sigma}(x, x')$, définie pour tout x, x' dans Σ par :

$$g_{\Sigma}(x, x') = v_{\Sigma}(x, x') \frac{\cos \theta' \cos \theta_{x'x}}{\|x - x'\|^2} .$$

L'angle solide élémentaire $d\sigma(w')$ s'écrit :

$$d\sigma(w') = \frac{\cos \theta_{x'x}}{\|x - x'\|^2} d\mu(x') .$$

Pour effectuer le changement de variable dans l'intégrale de l'équation 1.13, il est nécessaire d'introduire la fonction de visibilité de manière à pouvoir écrire l'intégrale sur un ensemble de surfaces en ne tenant compte que des points qui ont une contribution sur le résultat. Cette équation s'écrit alors :

$$\phi(x) = q_b(x) + \rho(x) \int_{\Sigma(x)} \phi(x') \cos \theta' v_{\Sigma}(x, x') \frac{\cos \theta_{x'x}}{\|x - x'\|^2} d\mu(x') , \quad \forall x \in \Sigma ,$$

où $\Sigma(x)$ est un sous-ensemble de Σ défini par :

$$\Sigma(x) = \{x' \in \Sigma , x' = \xi_{\Sigma}(x, -w') , w' \in \Omega(x)\} .$$

L'équation 1.13 s'écrit finalement en fonction du facteur géométrique de la manière suivante :

$$\phi(x) = q_b(x) + \rho(x) \int_{\Sigma(x)} \phi(x') g_{\Sigma}(x, x') d\mu(x') , \quad \forall x \in \Sigma . \quad (1.14)$$

1.2 Grandeurs radiométriques

1.2.1 Luminance

Un système optique est l'ensemble des organes d'un instrument qui vont de la production d'un rayonnement à sa détection. L'étude du comportement d'un système optique s'intéresse donc aux propriétés radiatives des sources, à la propagation dans les milieux et les composants optiques, et aux caractéristiques des détecteurs.

La radiométrie (Desvignes [33]) est la discipline qui concerne la caractérisation théorique et expérimentale des rayonnements optiques. La luminance est la grandeur radiométrique qui permet de caractériser la géométrie d'un rayonnement, c'est-à-dire sa répartition spatiale et angulaire. La luminance, notée $L(x, w)$, est une fonction définie sur $X \times \mathcal{S}^2$ à valeurs dans \mathbb{R} proportionnelle à la densité angulaire de flux $\phi(x, w)$:

$$\boxed{L(x, w) = E_\lambda \phi(x, w)} \quad [Wm^{-2}sr^{-1}] ,$$

où E_λ est l'énergie d'un photon de longueur d'onde λ .

De manière analogue, les fonctions d'émission volumique, notée $\varepsilon(x, w)$, et d'émission surfacique, notée $\varepsilon_b(x, w)$, sont proportionnelles aux densités angulaires de flux correspondantes :

$$\begin{cases} \varepsilon(x, w) &= E_\lambda q(x, w) & [Wm^{-3}sr^{-1}] \\ \varepsilon_b(x, w) &= E_\lambda q_b(x, w) & [Wm^{-2}sr^{-1}] \end{cases} .$$

L'équation de transfert en luminance se déduit de l'équation 1.3 en la multipliant par l'énergie du photon E_λ . Ainsi, pour tout $(x, w) \in (X \setminus \Sigma) \times \mathcal{S}^2$, c'est-à-dire pour tout point n'appartenant pas à l'ensemble des surfaces, cette équation s'écrit :

$$\boxed{\varepsilon(x, w) + \int_{\mathcal{S}^2} k(x, \langle w', w \rangle) L(x, w') d\sigma(w') = \langle w, \nabla_x L(x, w) \rangle + \sigma(x)L(x, w)} . \quad (1.15)$$

De même, la luminance au niveau d'une surface se déduit en multipliant l'équation 1.9 par E_λ . Elle s'écrit pour tout $(x, w) \in \Sigma \times \mathcal{S}^2$, c'est à dire pour tout point de l'ensemble des surfaces :

$$\boxed{L(x, w) = \varepsilon_b(x, w) + \int_{\mathcal{S}^2} k_b(x, w' \rightarrow w) L(x, w') d\sigma(w')} . \quad (1.16)$$

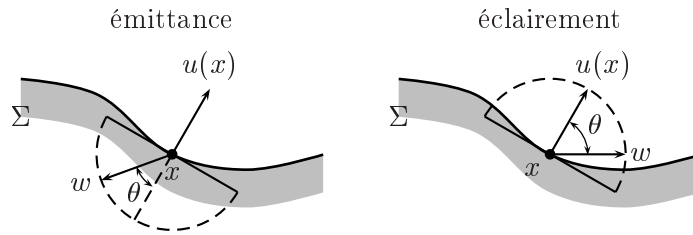
1.2.2 Émittance et éclairement

La densité de flux émis dans une demi-sphère est une fonction à valeurs dans \mathbb{R} , appelée émittance, notée $M(x)$, définie par :

$$M(x) = \int_{\Omega^+(x)} L(x, w) \cos \theta d\sigma(w) \quad [Wm^{-2}] .$$

La densité de flux reçu dans une demi-sphère est une fonction à valeurs dans \mathbb{R} , appelée éclairement, notée $E(x)$, définie par :

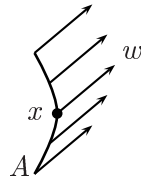
$$E(x) = \int_{\Omega^-(x)} L(x, w) \cos \theta d\sigma(w) \quad [Wm^{-2}] .$$

Fig. 1.18 – *Émittance et éclairement.*

1.2.3 Intensité optique

La densité de flux traversant une surface A est une fonction à valeurs dans \mathbb{R} , appelée intensité optique, notée $I_A(w)$, définie par :

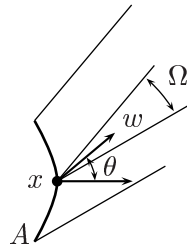
$$I_A(w) = \int_A L(x, w) d\mu(x) \quad [Wsr^{-1}] .$$

Fig. 1.19 – *Intensité optique à travers une surface.*

1.2.4 Flux

Le flux de photons à travers une surface A , se déplaçant dans une direction appartenant à un angle solide Ω est une quantité réelle donnée par :

$$\Phi_{A \times \Omega} = \int_{\Omega} \int_A L(x, w) \cos \theta d\mu(x) d\sigma(w) \quad [W] .$$

Fig. 1.20 – *Flux à travers une surface.*

La quantité $\cos \theta dw dx$ est appelée étendue géométrique élémentaire.

1.2.5 Noyau de dispersion surfacique

En milieu industriel, en particulier pour la radiométrie, les caractéristiques optiques des surfaces, qui sont le lieu des points de discontinuité de l'indice de réfraction, sont d'une importance fondamentale. Les propriétés de ces surfaces sont définies de manière plus ou moins précise selon les besoins de l'étude. La grandeur qui caractérise la propriété optique d'une surface est la BSSDF (*Bidirectional Surface-Scattering Distribution Function*) ou plus simplement BDF, notée $\rho(x, w' \rightarrow w)$, reliée au noyau de dispersion surfacique par la relation suivante, pour tout $(x, w) \in \Sigma \times \mathcal{S}^2$:

$$\begin{aligned} k_b(x, w' \rightarrow w) &= \rho(x, w' \rightarrow w) \cos \theta' \\ &= \rho(x, w' \rightarrow w) |\langle w', u(x) \rangle|, \quad \forall w' \in \mathcal{S}^2. \end{aligned}$$

Remarque

Le principe de conservation de l'énergie, première loi de la thermodynamique, précise que l'énergie interagissant avec la surface ne peut-être supérieure à l'énergie incidente :

$$\int_{\mathcal{S}^2} \rho(x, w' \rightarrow w) \cos \theta \, d\sigma(w) \leq 1, \quad \forall w' \in \mathcal{S}^2,$$

où $\cos \theta = |\langle w, u(x) \rangle|$.

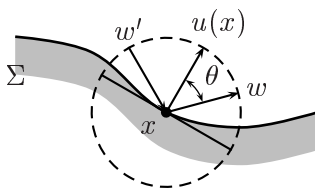


Fig. 1.21 – Principe de conservation de l'énergie.

□

Remarque

Le principe de réciprocité d'Helmholtz, deuxième loi de la thermodynamique, précise que :

$$\rho(x, w' \rightarrow w) = \rho(x, w \rightarrow w'), \quad \forall w \in \mathcal{S}^2, \quad \forall w' \in \mathcal{S}^2.$$

□

En pratique la BDF n'est pas connue exactement. Il existe de nombreuses modélisations des propriétés de surface (Glassner [44]). Généralement, la BDF est séparée en deux fonctions, l'une modélisant l'énergie réfléchiée par la surface (BRDF), et l'autre l'énergie transmise (BTDF).

Si $\Omega(x) = \Omega^-(x)$, alors le noyau de dispersion surfacique peut être relié à la BRDF (*Bidirectional Reflectance Distribution Function*), notée $\rho_r(x, w' \rightarrow w)$, par la relation suivante, pour tout $(x, w) \in \Sigma \times \Omega^+(x)$:

$$\begin{aligned} k_b(x, w' \rightarrow w) &= \rho_r(x, w' \rightarrow w) \cos \theta' \\ &= -\rho_r(x, w' \rightarrow w) \langle w', u(x) \rangle, \quad \forall w' \in \Omega^-(x). \end{aligned}$$

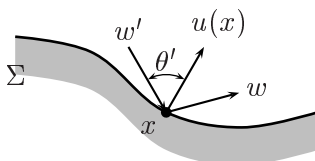


Fig. 1.22 – Géométrie de la BRDF.

De manière analogue, si $\Omega(x) = \Omega^+(x)$, alors le noyau de dispersion surfacique peut être relié à la BTDF (*Bidirectional Transmission Distribution Function*), notée $\rho_t(x, w' \rightarrow w)$, par la relation suivante, pour tout $(x, w) \in \Sigma \times \Omega^+(x)$:

$$\begin{aligned} k_b(x, w' \rightarrow w) &= \rho_t(x, w' \rightarrow w) \cos \theta' \\ &= \rho_t(x, w' \rightarrow w) \langle w', u(x) \rangle, \quad \forall w' \in \Omega^+(x). \end{aligned}$$

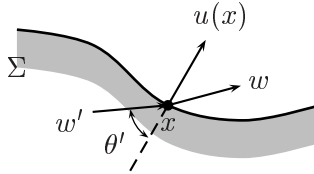


Fig. 1.23 – Géométrie de la BTDF.

Pour une surface donnée, la BDF est généralement issue de mesures expérimentales. Les directions angulaires w et w' sont échantillonnées en fonction de la précision voulue et des caractéristiques de l'appareil de mesure. La BDF est alors connue pour un certain nombre de valeurs ρ_k en les points $\{(x, w'_k \rightarrow w_k)\}_k$. Son expression peut être approchée dans une base de fonctions $\{\psi_j\}_j$ de la manière suivante :

$$\rho(x, w' \rightarrow w) = \sum_j a_j \psi_j(x, w' \rightarrow w),$$

où les coefficients $\{a_j\}_j$ sont déterminés à partir des valeurs mesurées $\{\rho_k\}_k$ et de la base de fonctions $\{\psi_j\}_j$. Cependant, dans certains cas particuliers, il est possible d'utiliser une expression analytique de la BDF.

Réflexion spéculaire

La réflexion spéculaire fournit une expression simple de la BRDF.

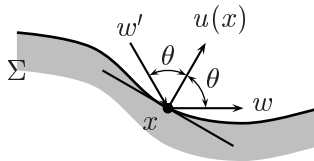


Fig. 1.24 – Géométrie de la réflexion spéculaire.

Les lois de Snell-Descartes précisent que les vecteurs w , w' et $u(x)$ sont coplanaires et que les angles d'incidence θ et de réflexion θ' sont égaux. La relation dans le triangle formé par ces trois vecteurs permet d'écrire :

$$\begin{aligned} w - w' &= 2 \langle w, u(x) \rangle u(x) \\ &= 2 \cos \theta u(x). \end{aligned}$$

La BRDF s'écrit alors de la manière suivante :

$$\rho_r(x, w' \rightarrow w) = \begin{cases} 1 & \text{si } w' = w - 2 \cos \theta u(x) \\ 0 & \text{sinon} \end{cases}.$$

La densité angulaire de flux au niveau d'une surface (équation 1.9) s'écrit, en tout point $(x, w) \in \Sigma \times \mathcal{S}^2$ où il y a réflexion spéculaire, de la manière suivante :

$$\phi(x, w) = q_b(x, w) + \phi(x, w - 2 \cos \theta u(x)) \cos \theta .$$

Transmission spéculaire

Le cas de la transmission spéculaire est analogue.

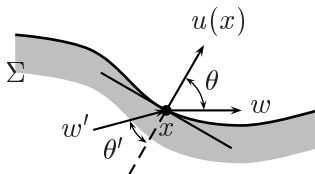


Fig. 1.25 – Géométrie de la transmission spéculaire.

Pour exprimer la relation entre les vecteurs w et w' il est nécessaire d'introduire les indices de réfraction de chaque côté de la surface Σ . Soit $\eta^+(x)$ l'indice de réfraction situé du côté du vecteur normal $u(x)$ et $\eta^-(x)$ l'indice de réfraction situé de l'autre côté. Les lois de Snell-Descartes précisent que les vecteurs w , w' et $u(x)$ sont coplanaires et que les angles θ et θ' sont reliés par :

$$\eta^+(x) \sin \theta = \eta^-(x) \sin \theta' .$$

La relation entre les différents vecteurs coplanaires s'écrit alors :

$$w' = \zeta(x) w - \left(\zeta(x) \cos \theta - \sqrt{1 - \zeta^2(x) \sin^2 \theta} \right) u(x) ,$$

où $\zeta(x)$ est le rapport des indices de réfraction :

$$\zeta(x) = \frac{\eta^+(x)}{\eta^-(x)} .$$

La BTDF s'écrit alors de la manière suivante :

$$\rho_t(x, w' \rightarrow w) = \begin{cases} 1 & \text{si } w' = \zeta(x) w - \left(\zeta(x) \cos \theta - \sqrt{1 - \zeta^2(x) \sin^2 \theta} \right) u(x) \\ 0 & \text{sinon} \end{cases} .$$

La densité angulaire de flux au niveau d'une surface (équation 1.9) s'écrit, pour tout point $(x, w) \in \Sigma \times \mathcal{S}^2$ où il y a transmission spéculaire, de la manière suivante :

$$\phi(x, w) = q_b(x, w) + \phi \left(x, \zeta(x) w - \left(\zeta(x) \cos \theta - \sqrt{1 - \zeta^2(x) \sin^2 \theta} \right) u(x) \right) \sqrt{1 - \zeta^2(x) \sin^2 \theta} .$$

Réflexion diffuse

La réflexion diffuse est un cas encore plus simple. Il correspond à la propriété optique dite lambertienne d'une surface.

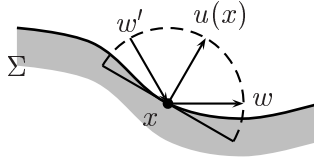


Fig. 1.26 – Géométrie de la réflexion diffuse.

La BRDF est constante quelles que soient les directions w et w' considérées :

$$\rho_r(x, w' \rightarrow w) = \rho_r(x) .$$

L'indépendance de la BRDF par rapport à la direction implique une indépendance de la densité angulaire de flux par rapport à la direction pour les points appartenant à l'ensemble des surfaces. Ainsi, la densité angulaire de flux au niveau d'une surface (équation 1.9) s'écrit, pour tout point $x \in \Sigma$ où il y a réflexion diffuse, de manière analogue à l'équation 1.14 :

$$\phi(x) = q_b(x) + \rho_r(x) \int_{\Sigma^+(x)} \phi(x') g(x, x') d\mu(x') ,$$

où $\Sigma^+(x)$ est un sous-ensemble de Σ défini par :

$$\Sigma^+(x) = \{x' \in \Sigma , x' = \xi_\Sigma(x, -w') , w' \in \Omega^+(x)\} .$$

Transmission diffuse

De manière identique, la transmission diffuse précisant que la BTDF est indépendante des directions w et w' permet d'écrire la densité angulaire de flux au niveau d'une surface (équation 1.9) de la manière suivante :

$$\phi(x) = q_b(x) + \rho_t(x) \int_{\Sigma^-(x)} \phi(x') g(x, x') d\mu(x') ,$$

où $\Sigma^-(x)$ est un sous-ensemble de Σ défini par :

$$\Sigma^-(x) = \{x' \in \Sigma , x' = \xi_\Sigma(x, -w') , w' \in \Omega^-(x)\} .$$

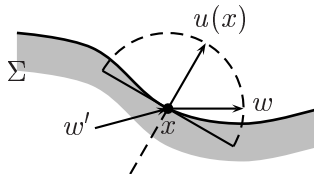


Fig. 1.27 – Géométrie de la transmission diffuse.

1.3 Conclusion

Ce chapitre donne une idée précise du modèle physique considéré dans ce projet industriel. Le modèle du transport est suffisamment consistant pour décrire les phénomènes qui intéressent les opticiens dans ce contexte de travail. Lorsque le milieu considéré est homogène par morceaux alors l'équation

de Boltzmann accompagnée des conditions de bord sur les surfaces se ramène à l'équation de radiativité introduite par Kajiyama [52] qui fait encore référence aujourd'hui dans le domaine de l'image de synthèse. Différents problèmes subsistent quant à l'utilisation en simulation de ce modèle du transport. En particulier, la connaissance du noyau de dispersion surfacique, qui est un des paramètres fondamentaux, est un problème difficile, donnant lieu à un nombre croissant de travaux. Cependant, les opticiens ont bien souvent une idée du profil de ce noyau en fonction des matériaux des surfaces considérées. Ainsi, à partir d'un échantillon expérimental de petite taille, on sait décrire un noyau de dispersion surfacique suffisamment précis par rapport au besoin.

Dans la suite, on se placera dans des milieux homogènes par morceaux et on supposera disposer de tous les paramètres nécessaires à la recherche d'une solution pour ce modèle.

Chapitre 2

Représentation des objets et algorithmes de base

Un système optique est défini par une géométrie et des caractéristiques physiques. La géométrie permet de représenter les surfaces qui sont le lieu des points de discontinuité de l'indice de réfraction optique. La représentation de ces surfaces peut se faire à l'aide de différents modèles géométriques. Si, dans un premier temps, les opticiens ont considéré, pour la simulation physique des phénomènes radiométriques, des formes géométriques simples (lentille, sphère, cylindre, etc), les contraintes industrielles et les domaines d'applications imposent désormais une définition des objets à considérer selon des normes de la CAO (surfaces paramétrées principalement). Cet historique est d'ailleurs conforme à ce qui s'est passé dans le domaine de la mécanique et des techniques informatiques associées.

Ainsi, dans l'entreprise qui a constitué le contexte de ce travail, le logiciel a dû évoluer d'un système fondé sur un modèle géométrique CSG (section 2.2) vers un système comprenant en plus un modèle géométrique BRep (section 2.3). L'objet principal de ce travail étant d'accompagner cette évolution, nous avons dû étudier les algorithmes d'intersection rayon-objet pour ces deux modèles. Dans le premier paragraphe (section 2.1), on introduit les notions de base relatives à l'étude des complexités, nécessaires pour mener à bien l'analyse des algorithmes étudiés.

Tout au long de ce chapitre, nous avons essayé de rédiger pédagogiquement et exhaustivement les notions de base, de manière à constituer un document introductif sur le sujet pour l'entreprise concernée.

2.1 Complexités

L'étude des complexités en mémoire et en temps est nécessaire pour, d'une part, analyser et comprendre les structures de données et les algorithmes mis en œuvre pour les méthodes considérées, et, d'autre part, tirer de cette analyse des informations utiles à la recherche de nouvelles méthodes et techniques pour atteindre le but fixé, à savoir l'accélération des calculs d'intersection. Bien que la complexité en temps soit la préoccupation principale de ce travail, il ne faut pas négliger la complexité en mémoire, qui, dans le cadre industriel, doit rester acceptable au vu du matériel sur lequel les simulations sont effectuées.

Dans le contexte qui nous intéresse, le travail à réaliser doit s'intégrer dans un environnement existant. Les structures de données définissant la géométrie et les propriétés physiques des objets et les algorithmes propres à la simulation sont fixés. C'est pourquoi on commencera par étudier les complexités relatives à l'environnement existant, de manière à cadrer le travail d'amélioration des performances.

Les notions de complexité en mémoire d'une structure de données ou de complexité en temps d'un algorithme sont définies dans de nombreux ouvrages informatiques généraux¹.

¹Voir des auteurs tels que Froidevaux, Gaudel et Soria [40] ou Cormen, Leiserson et Rivert [23].

2.1.1 Complexité en mémoire

Dans le contexte industriel, les complexités en mémoire concernent toutes les informations nécessaires à la définition d'un système optique, principalement la géométrie et les propriétés physiques, mais aussi des informations nécessaires aux calculs d'une simulation. Dans ce travail, nous ne tiendrons compte que de la géométrie du système optique et des structures de données éventuellement nécessaires aux calculs. De manière générale, la complexité en mémoire est un nombre, que l'on note M , et que l'on indice par l'entité considéré. Par exemple, pour un objet O , la complexité en mémoire associée est notée M_O . Pour l'évaluation de ces complexités on considérera qu'une unité de complexité en mémoire correspond à la place occupée par un nombre entier ou réel.

2.1.2 Complexité en temps

Les complexités en temps proviennent de l'ensemble des algorithmes utilisés lors d'une simulation. Dans le cadre de ce travail, ce sont les calculs d'intersection entre des rayons et le système optique considéré qui nous intéressent. La complexité en temps de ces calculs représente une partie prépondérante dans une simulation, ce qui donne un sens à cette étude ciblée en vue de la réduction des temps de simulation. La justification de cette remarque est faite par la suite.

La complexité en temps de recherche de la première intersection entre un rayon et un système optique est difficile à écrire. En effet, l'ensemble des rayons qui seront traités lors d'une simulation ne sont pas connus a priori. Il faut, en plus de la géométrie, tenir compte des propriétés optiques des objets pour prendre en compte les émissions de photons et les interactions entre photons et objets. Ainsi, la complexité en temps dépend des rayons considérés, de la géométrie et des propriétés optiques. Notre travail est centré sur la partie géométrique, par conséquent il faut trouver un moyen d'évaluer la complexité en temps sans se préoccuper des propriétés optiques.

Pour un système optique donné (géométrie et propriétés optiques), dans un domaine X , de très grandes quantités de rayons sont considérées pour une simulation. Soit \mathcal{R} l'ensemble des couples (x, w) caractérisant tous les rayons pouvant être considérés pour un système optique donné :

$$\mathcal{R} \subset X \times \mathcal{S}^2 .$$

\mathcal{R} est donc un sous-ensemble de l'espace $\mathbb{R}^3 \times \mathcal{S}^2$, appelé espace des phases. On montre que l'on peut définir une mesure ϱ sur l'espace des phases (Arvo [3]), construite à partir des mesures de Lebesgue sur X et \mathcal{S}^2 , notées respectivement μ et σ . On définit alors la mesure produit ϱ par complétion sur l'espace produit $\mathbb{R}^3 \times \mathcal{S}^2$ et on l'écrit de la manière suivante² :

$$\varrho = \mu \times \sigma .$$

\mathcal{R} est donc un espace ϱ -mesurable, de mesure finie (X est borné). On note $\text{mes}(\mathcal{R})$ la mesure de cet espace que l'on écrit sous la forme intégrale suivante :

$$\text{mes}(\mathcal{R}) = \int_{\mathcal{R}} d\varrho(x, w) = \int_{\mathcal{R}} d\mu(x) d\sigma(w) .$$

La forme différentielle $d\mu(x)$ représente soit une surface élémentaire, soit un volume élémentaire autour de x selon que les origines des rayons, et donc des photons, se situent sur l'ensemble Σ des surfaces optiques ou non. La forme différentielle $d\sigma(w)$ est l'aire de la portion de sphère unité interceptée par un angle solide élémentaire autour de w . La forme différentielle $d\varrho(x, w)$ est donc d'ordre 4 ou 5.

De manière générale, on définit T_O la fonction de complexité en temps d'un algorithme de calcul d'intersection entre un rayon et un objet O du système optique considéré :

$$\begin{aligned} T_O : \quad \mathcal{R} &\longrightarrow \mathbb{R}^+ \\ (x, w) &\longmapsto T_O(x, w) \end{aligned} .$$

$T_O(x, w)$ est la complexité en temps du calcul d'intersection entre le rayon $R_{x,w}$ et l'objet O .

²Voir des ouvrages généraux, tel que celui écrit par Rubin [71]

Propriétés de T_O

\mathcal{R} est un ensemble mesurable de mesure non nulle et finie. C'est une constatation qui a bien un sens physique (continuité des objets et des milieux considérés). Les notions classiques de l'analyse peuvent être utilisées pour les fonctions définies sur \mathcal{R} . Une propriété triviale de T_O est que c'est une fonction bornée sur \mathcal{R} :

$$0 < T_O(x, w) < +\infty, \forall (x, w) \in \mathcal{R}.$$

Puisque \mathcal{R} est borné, cela implique que T_O est une fonction de l'espace métrique $L^1(\mathcal{R})$.

La caractéristique de cette fonction qui nous intéresse, et qui sera utile tout au long de ce travail, est sa nature, et en particulier sa régularité. Les algorithmes intervenant dans l'évaluation de cette fonction n'assurent pas que T_O est continue sur \mathcal{R} . Par contre, on peut faire l'hypothèse que T_O est constante par morceaux sur \mathcal{R} . Autrement dit, il existe une partition $\{\mathcal{R}_1, \dots, \mathcal{R}_{n_r}\}$ de \mathcal{R} telle que T_O est constante sur chaque \mathcal{R}_j :

$$T_O(x, w) = T_O|_{\mathcal{R}_j}, \forall (x, w) \in \mathcal{R}_j.$$

Les méthodes de calcul d'intersection rayon-objet sont de deux types. Soit l'équation paramétrique du rayon est injectée dans le système d'équations et/ou d'inéquations, puis le système obtenu, dont l'inconnue est le paramètre du rayon, est résolu. Soit l'objet considéré est approché par des objets simples (des faces planes par exemple), puis l'intersection entre ces objets simples et le rayon est recherchée par des méthodes géométriques simples (intersection rayon-plan par exemple), et, éventuellement, une projection de la solution sur l'objet est effectuée. La plupart des algorithmes utilisent des critères géométriques d'exclusion (objets englobants par exemple), qui consistent à éliminer les cas triviaux où il n'y a pas de solution.

Dire que T_O est localement constante signifie que pour des rayons proches le comportement algorithmique des méthodes envisagées est semblable. Plusieurs remarques renforcent cette constatation. Tout d'abord, les objets considérés sont réguliers, en ce sens que leur contour est localement continu (on ne considère pas des objets ponctuels ou linéiques). Ainsi, pour des rayons proches, les systèmes à résoudre sont semblables et engendrent des solutions proches, et, pour des approximations de l'objet considéré, les méthodes géométriques envisagées se comportent de manières similaires.

T_O est définie par morceaux car des discontinuités apparaissent pour des rayons tangents à l'objet considéré (passage d'existence d'une solution à aucune solution), ou pour des rayons se trouvant dans des parties de l'objet avec une variation importante de la solution (variation du nombre de solutions), ou encore pour des rayons se situant au niveau d'un critère géométrique d'exclusion.

Remarque

Soit λ_O la fonction représentant le nombre d'intersections entre un rayon issu de \mathcal{R} et l'objet O :

$$\begin{aligned} \lambda_O : \quad \mathcal{R} &\longrightarrow \mathbb{N} \\ (x, w) &\longmapsto \lambda_O(x, w) \end{aligned}$$

Les propriétés de λ_O sont semblables à celles de T_O , pour les mêmes raisons. C'est une fonction de $L^1(\mathcal{R})$, elle est constante par morceaux. Mais, on doit aussi remarquer que la partition de \mathcal{R} où T_O est constante est une sous-partition de celle où λ_O est constante. Cela vient du fait que le nombre d'intersections influe nécessairement sur la complexité en temps du calcul d'intersection puisque les discontinuités de T_O sont dues, entre autres, au changement du nombre de solutions. C'est une sous-partition car les critères d'exclusions géométriques font que pour un même nombre de solutions la complexité en temps T_O n'est pas constante. Cette propriété peut s'écrire de la manière suivante :

$$T_O \text{ constante sur } \mathcal{R}_j \implies \lambda_O \text{ constante sur } \mathcal{R}_j,$$

où \mathcal{R}_j est un sous-ensemble de \mathcal{R} . Cette propriété remarquable est illustrée dans les expérimentations qui suivent.

□

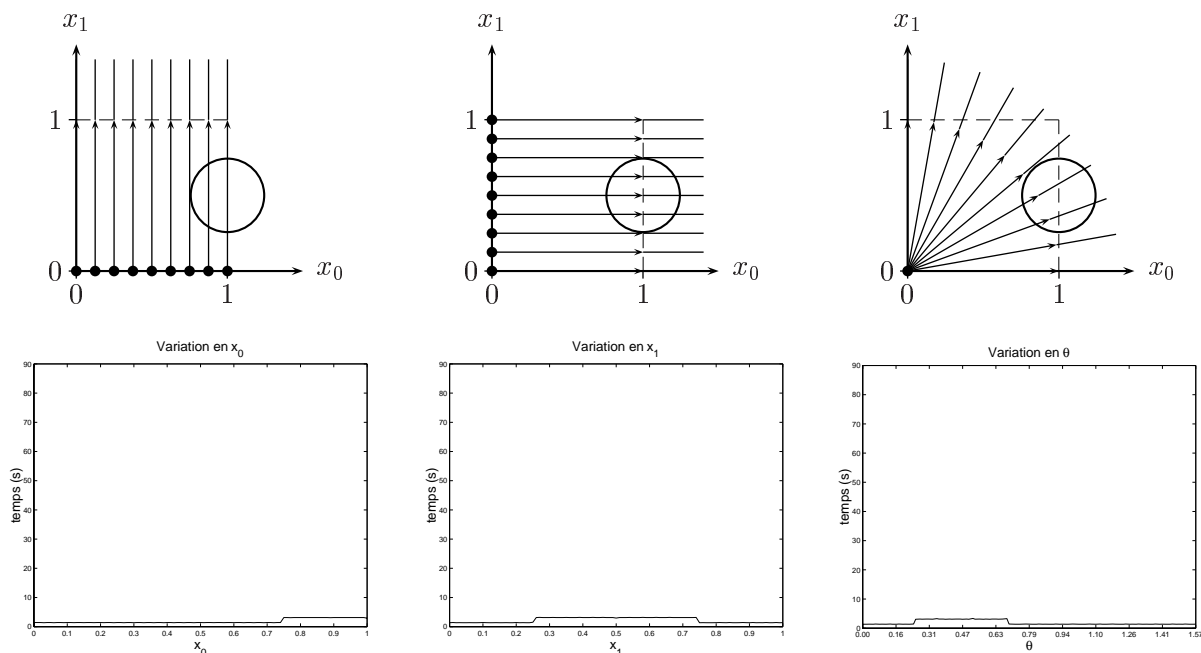
Expérimentations

Pour illustrer cette hypothèse sur la régularité de la fonction T_O , considérons trois exemples en dimension 2. Le cas de la dimension 3 est identique, mais il est plus simple de mettre en place des exemples en dimension 2 car il y a moins de paramètres à considérer. Pour ce faire, considérons un objet, et faisons varier les paramètres qui définissent le rayon, c'est-à-dire les variables x et w . Plus précisément, on fait varier les deux variables d'espace x_0 et x_1 et la variable d'orientation θ qui est l'angle entre l'axe x_0 est w . Pour chaque rayon considéré, un grand nombre de calculs d'intersection rayon-objet est effectué (échantillon de taille 1000 000). On obtient ainsi une valeur expérimentale représentative. Les différentes valeurs obtenues sont représentées sur un graphique, qui donne les variations de la fonction T_O par rapport à la variable prise en compte. Trois ensembles de rayons sont pris en compte :

- variation en x_0 : $\mathcal{R} = ([0, 1] \times \{0\}) \times \{(0, 1)\}$,
- variation en x_1 : $\mathcal{R} = (\{0\} \times [0, 1]) \times \{(1, 0)\}$,
- variation en θ : $\mathcal{R} = \{(0, 0)\} \times \{(w_0, w_1) \in \mathcal{S}^1, \theta = \arccos w_0 \in [0, \pi/2]\}$.

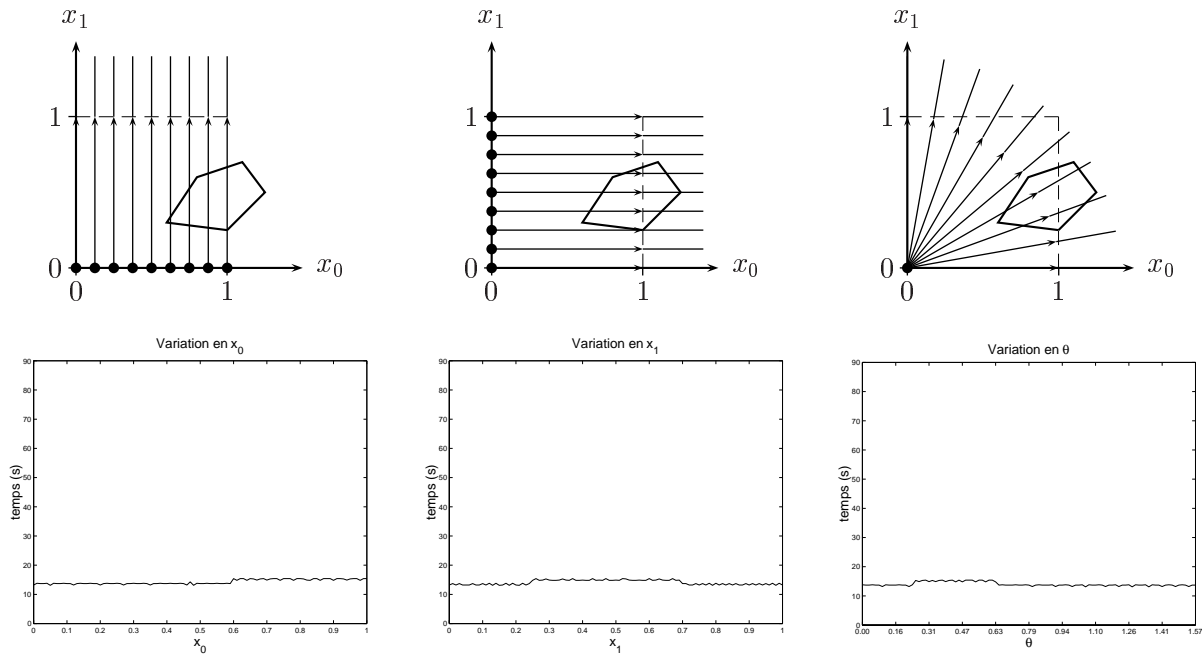
Ces expérimentations ont pour but de donner une idée de la forme des courbes des fonctions T_O . Il n'est donc pas nécessaire d'accorder de l'importance aux valeurs de ces fonctions aux différents points considérés. De plus, ces valeurs n'ont de sens que pour un matériel et un environnement de développement précis, et ne sont donc pas significatives. Par contre, la forme des courbes dépend des méthodes et des algorithmes utilisés, et reste donc significative quel que soit le matériel et l'environnement de développement considérés.

Pour le premier exemple, l'objet considéré est un cercle. La méthode de recherche d'intersection consiste à injecter l'équation du rayon dans l'équation implicite du cercle puis à résoudre une équation du second degré en le paramètre du rayon. Le graphique montre bien la différence de temps de calcul entre des rayons qui ont une intersection avec le cercle et des rayons qui n'en ont pas.



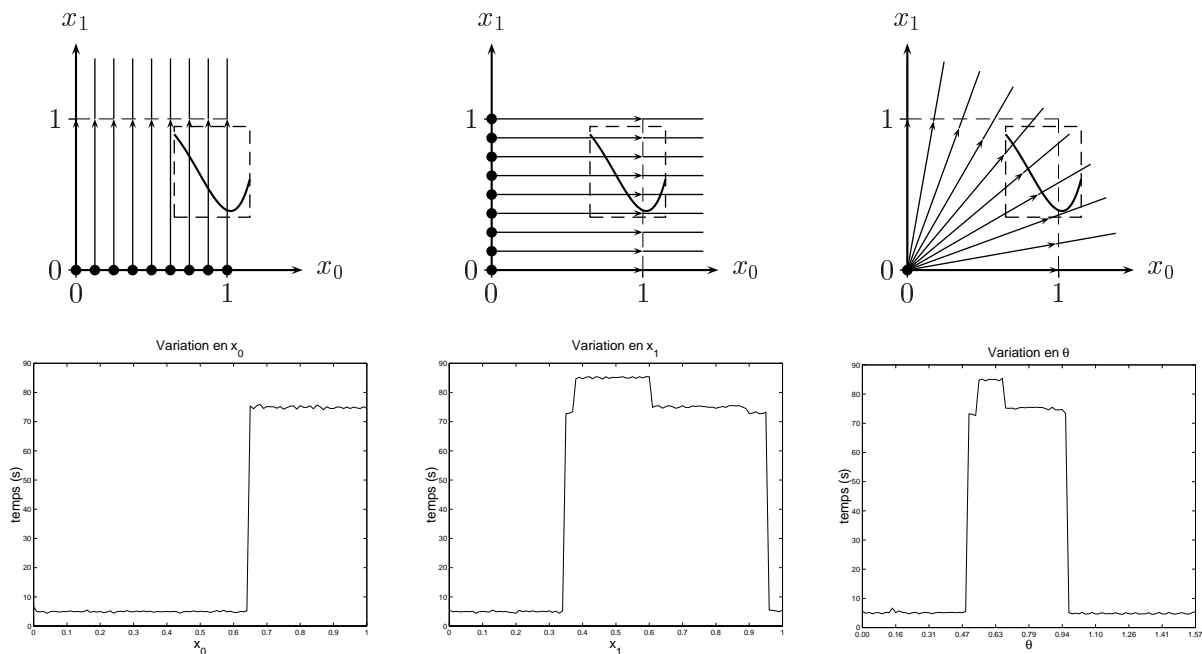
Variations du temps de calcul d'intersection entre un rayon et un cercle.

Dans le deuxième exemple, l'objet considéré est un polygone. La méthode de recherche d'intersection consiste à calculer direction l'intersection entre le rayon et chaque segment. La même remarque que dans le cas du cercle peut être faite.



Variations du temps de calcul d'intersection entre un rayon et un polygone.

Dans le troisième exemple, l'objet considéré est un polynôme. La méthode de résolution consiste tout d'abord à tester l'intersection entre le rayon et un rectangle englobant le polynôme, et, s'il y a intersection, à injecter l'équation du rayon dans le polynôme, puis à isoler les racines du polynôme obtenu, enfin à affiner la précision à l'aide d'une méthode de Newton. Le graphique montre bien les discontinuités dues au critère géométrique d'exclusion par le rectangle englobant, et aussi les discontinuités dues au nombre de racines trouvées.



Variations du temps de calcul d'intersection entre un rayon et un polynôme.

Complexité moyenne en temps

Ce paragraphe a pour but de donner une définition de la complexité en temps de calcul d'intersection entre un rayon et un objet moyennée par rapport à l'ensemble des rayons issus de \mathcal{R} .

Définition 1

La complexité moyenne en temps de la fonction T_O par rapport à l'ensemble \mathcal{R} est définie de la manière suivante :

$$\bar{T}_O = \frac{1}{\text{mes}(\mathcal{R})} \int_{\mathcal{R}} T_O(x, w) d\mu(x) d\sigma(w) . \quad (2.1)$$

□

Cette définition est valable si l'intégrale existe. \mathcal{R} est borné et la fonction T_O est dans $L^1(\mathcal{R})$, donc l'intégrale a bien un sens. L'intérêt de cette approche réside dans le fait que l'on définit une expression de la complexité relatant l'ensemble des rayons considérés pour une simulation donnée. Cette complexité moyenne en temps est donc représentative de la simulation considérée. Une des principales difficultés est de définir l'ensemble \mathcal{R} . Nous verrons au chapitre 4 un moyen de définir un ensemble représentatif des simulations considérées.

2.2 Modèle géométrique CSG

Le modèle géométrique de type CSG (*Constructive Solid Geometry*) définit une représentation constructive des objets. Un objet est décrit par un arbre de construction, les nœuds de cet arbre portent des opérateurs géométriques ou ensemblistes, les feuilles sont des objets primitifs. Des informations supplémentaires peuvent être adjointes aux feuilles et aux nœuds (propriétés optiques, boîtes englobantes, etc).

Dans un modèle CSG, la représentation par un arbre de construction d'un objet fournit une description implicite en ce sens que le contour de l'objet n'est pas connu explicitement, il faut l'évaluer. De manière générale, cette description pose certains problèmes. Par exemple, il est difficile de tester si un objet décrit l'objet vide. Aussi, il est nécessaire de s'assurer de la cohérence des résultats par application des opérateurs géométriques et ensemblistes. Cependant, ce modèle est basé sur la théorie des ensembles, il est par conséquent fiable et non ambigu.

2.2.1 Description

Requicha et Voelcker [67] furent les premiers à définir un modèle géométrique CSG pour la représentation d'objets solides dans le contexte de la mécanique. De façon succincte, un modèle de type CSG est défini par :

- un ensemble d'objets primitifs $\{P_i\}_i$,
- un ensemble d'opérateurs géométriques $\{G_i\}_i$,
- un ensemble d'opérateurs ensemblistes $\{E_i\}_i$.

Un objet O relevant de ce modèle est obtenu par applications successives d'opérateurs géométriques et/ou ensemblistes sur des objets primitifs ou des objets du modèle. C'est pourquoi il peut être écrit de la manière suivante :

$$O = \begin{cases} P & \text{si c'est un objet primitif} \\ (\mathcal{O}_p, O_1, \dots, O_n) & \text{sinon} \end{cases} ,$$

où \mathcal{O}_p est un opérateur n -aire et O_1, \dots, O_n sont des objets CSG. Un objet O de type CSG, non primitif, est donc représentable par un arbre syntaxique, appelé arbre CSG (figure 2.1).

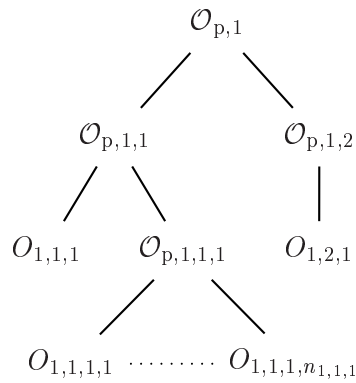


Fig. 2.1 – Représentation par un arbre syntaxique d'un objet de type CSG.

Pour décrire un système il est nécessaire de définir plusieurs objets. C'est pourquoi on introduit la notion de scène géométrique qui est une liste (ou collection) d'objets de type CSG. On notera O^1, \dots, O^N une telle liste composée de N objets.

Objets primitifs

La mise en œuvre d'un modèle CSG dépend beaucoup de la définition des objets primitifs. Leur nature répond aux besoins du contexte. Dans le cadre industriel de l'optique une importance particulière est accordée aux surfaces³. Les objets primitifs doivent permettre de décrire un changement de milieu, c'est pourquoi ce sont des objets fermés. Ainsi l'information relative aux propriétés optiques peut être portée par l'objet. Il suffit de définir deux indices de réfraction optique⁴ (intérieur et extérieur de l'objet). Certains objets primitifs, tel que la lentille, sont spécifiques au domaine (figure 2.2).

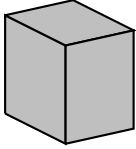
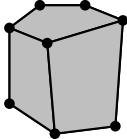






pavé	polyèdre fermé	sphère	ellipsoïde
			
3 longueurs	structure FCVL	rayon	diamètres
cylindre	cône	tore	lentille
			
rayon longueur	hauteur grand rayon demi-angle au sommet	petit et grand rayons	épaisseur diamètre extérieur rayons de courbure

Fig. 2.2 – Objets primitifs volumiques.

³Il faut comprendre le terme surface comme étant le lieu des points de discontinuité de l'indice de réfraction optique (voir page 15).

⁴Voir définition page 15.

Opérateurs géométriques

Les opérateurs géométriques permettent de définir des transformations géométriques à partir d'objets CSG. Les trois transformations de base sont :

- translation,
- rotation,
- homothétie.

Ces opérateurs sont de type unaire, par conséquent un objet résultant d'une transformation géométrique s'écrit :

$$O = (G, O_1) .$$

Pour décrire de telles transformations il est très pratique, d'un point de vue numérique, de se placer dans l'espace projectif, c'est-à-dire de travailler avec un point x de coordonnées $(x_0, x_1, x_2, 1)$ dans \mathbb{R}^4 . Ainsi chaque transformation est décrite par une matrice $(4, 4)$:

$$\begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ 1 \end{bmatrix} .$$

La translation de vecteur (v_0, v_1, v_2) est décrite par la matrice suivante :

$$\begin{bmatrix} 1 & 0 & 0 & v_0 \\ 0 & 1 & 0 & v_1 \\ 0 & 0 & 1 & v_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

Pour les rotations de centre $(0, 0, 0)$ et d'axe l'un des trois axes du repère, l'écriture est la suivante (rotation d'angle θ respectivement par rapport aux axes x_0, x_1 et x_2) :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

Pour une rotation de centre et d'axe de rotation quelconque il est nécessaire d'effectuer une translation puis une rotation pour se retrouver dans un repère correspondant à l'une des trois rotations de base, puis d'appliquer la rotation, enfin il faut se ramener dans le repère d'origine par une rotation inverse et une translation inverse. Cela se traduit par des produits de matrices.

L'homothétie de centre $(0, 0, 0)$ et de rapports dans chaque direction (s_0, s_1, s_2) est décrite par la matrice suivante :

$$\begin{bmatrix} s_0 & 0 & 0 & 0 \\ 0 & s_1 & 0 & 0 \\ 0 & 0 & s_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

Pour une homothétie de centre quelconque il est nécessaire d'effectuer une translation pour se placer dans le repère centré en le centre de l'homothétie, puis l'homothétie, et, enfin, la translation inverse pour se ramener au repère d'origine. Cela se traduit par des produits de matrices.

Toute autre transformation (telle que la réflexion ou la similitude) peut être décrite à partir de ces trois transformations de bases par des opérations de composition qui se traduisent par des produits

de matrices. Une liste d'opérateurs géométriques se ramène donc à une matrice $(4, 4)$ représentant la succession des transformations appliquées. Au niveau de la structure de données, pour connaître l'historique de la construction, il est nécessaire de conserver les paramètres de la succession des transformations géométriques appliquée à un objet. Cependant, la matrice correspondante pourra être calculée a priori.

Remarque

Dire qu'un opérateur géométrique est appliqué à un objet signifie qu'il est appliqué à l'ensemble des points constituant cet objet. Ce qui revient à effectuer une certaine transformation sur les paramètres définissant l'objet en question. Par exemple, appliquer une translation à une sphère revient à appliquer la translation à son centre. Appliquer une translation à une demi-droite (un rayon) décrite par une origine et une direction revient à appliquer la translation à son origine. Appliquer une rotation à une demi-droite revient à appliquer la rotation à l'origine et à la direction. Ce dernier exemple met en évidence que les transformations géométriques envisagées sur un objet donné ne s'appliquent pas telles quelles sur les paramètres définissant l'objet considéré.

Les transformations géométriques décrites dans ce paragraphe le sont pour des points de \mathbb{R}^3 , il faut donc garder à l'esprit que des transformations adéquates pour les éléments de \mathcal{S}^2 doivent être envisagées, de manière à pouvoir décrire, en particulier, la transformation géométrique associée à un rayon. \square

Opérateurs ensemblistes

Les opérateurs ensemblistes permettent de définir des objets par construction à partir d'objets primitifs ou d'objets de type CSG. Trois opérateurs de base suffisent à définir toutes les opérations envisageables :

- l'union (**ou**),
- l'intersection (**et**),
- la négation (**non**),

où l'on a noté entre parenthèses l'opération booléenne correspondante. Les trois opérateurs ensemblistes généralement utilisés sont l'union (**ou**), l'intersection (**et**) et la différence (**diff** = **et non**) (figure 2.3). La cohérence des objets résultants d'une construction à l'aide des opérateurs ensemblistes n'est pas systématiquement vérifiée de manière algorithmique. Il est donc nécessaire de respecter certaines contraintes lors de l'utilisation de ces opérateurs.

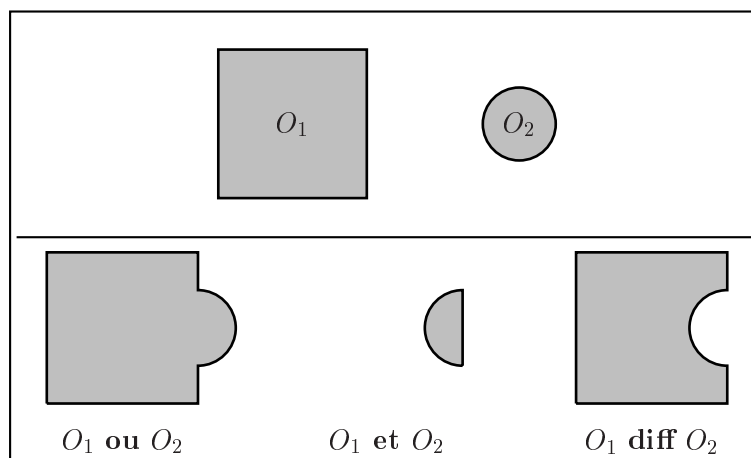


Fig. 2.3 – Définition des opérateurs ensemblistes.

2.2.2 Utilisation du modèle pour l'optique

Dans le contexte qui nous intéresse, la mise en place d'une simulation nécessite la définition d'un système optique, c'est-à-dire la représentation de la fonction indice de réfraction optique $\eta : X \subset \mathbb{R}^3 \rightarrow \mathbb{R}$. Ainsi, le contour d'un objet CSG est le lieu des points de discontinuité de la fonction η . Dans notre contexte, les objets peuvent être classés en deux catégories par rapport aux propriétés optiques :

- les objets susceptibles d'émettre des photons,
- les objets qui définissent une discontinuité de la fonction η .

Ainsi, un objet appartient à l'une, l'autre ou les deux catégories.

Objets primitifs

Les besoins particuliers de l'optique font qu'il est nécessaire, en plus des objets primitifs volumiques (figure 2.2), de définir des objets primitifs linéiques (figure 2.4) et des objets primitifs surfaciques (figure 2.5).

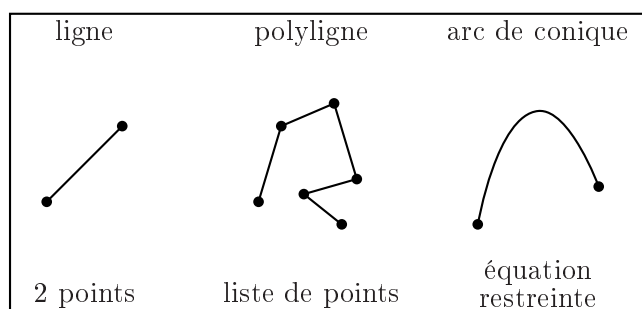


Fig. 2.4 – Objets primitifs linéiques.

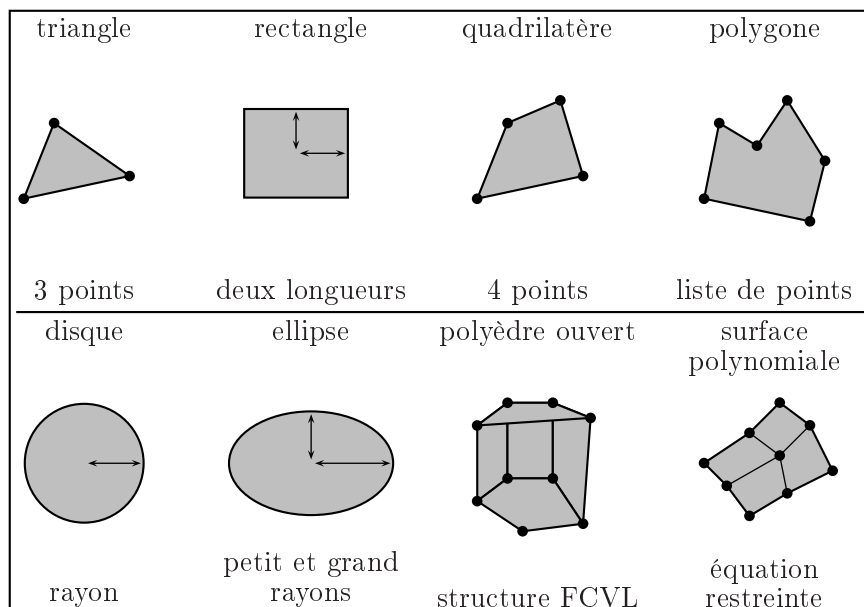


Fig. 2.5 – Objets primitifs surfaciques.

Les objets linéiques ne sont utilisés que pour la modélisation des sources de lumière telles que les filaments des ampoules. L'indice de réfraction optique d'un objet linéique est celui du milieu dans

lequel il est placé. En d'autres termes, il n'interagit pas avec des photons au cours d'une simulation. Par contre, les objets surfaciques ou volumiques définissent bien une discontinuité de la fonction η . Seuls les objets volumiques permettent de définir des milieux distincts, c'est-à-dire des valeurs de η distinctes. En général η est une constante dans un milieu donné.

Les objets primitifs sont caractérisés par un ensemble de paramètres. Comme nous le verrons par la suite, les calculs d'intersection entre des demi-droites et des surfaces constituent le coeur d'une simulation. Ces calculs doivent être précis et rapides. Une représentation par des fonctions implicites des objets primitifs est un choix judicieux dans ce contexte.

Enfin, notons que les objets primitifs linéiques et surfaciques ne peuvent être utilisés avec des opérateurs ensemblistes.

Opérateurs constructifs

Les opérateurs constructifs permettent de construire des objets à partir d'objets primitifs non nécessairement volumiques. Dans le cadre industriel de l'optique, deux opérateurs constructifs sont envisagés. Ils permettent de définir les entités géométriques suivantes :

- la surface de révolution, obtenue par rotation autour d'un axe d'une courbe 2D (assemblage de lignes et d'arcs de conique) (figure 2.6),
- la surface extrudée, obtenue par déplacement d'une courbe 2D (assemblage de lignes et d'arcs de conique) le long d'un segment (figure 2.6).

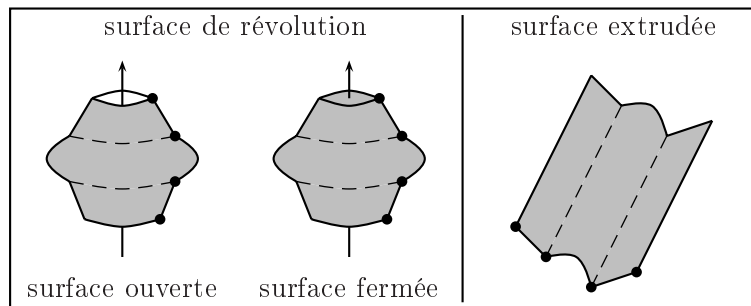


Fig. 2.6 – Opérateurs constructifs : surface de révolution et surface extrudée.

Une surface de révolution est une collection d'objets primitifs surfaciques, c'est par conséquent un objet surfacique. Cependant, il est possible de la considérer comme un objet volumique en rajoutant deux disques aux extrémités de la surface (figure 2.6). Dans ce dernier cas une surface de révolution peut être utilisée pour les opérateurs ensemblistes.

Une surface extrudée est une collection d'objets primitifs surfaciques, et c'est un objet surfacique si la courbe 2D considérée est ouverte, un objet volumique si elle est fermée (considérant que les domaines délimités par les courbes fermées aux extrémités sont pris en compte). Dans ce dernier cas, une surface extrudée peut être utilisée pour les opérateurs ensemblistes.

Un opérateur constructif est caractérisé par un ensemble de paramètres. Pour assurer la cohérence de l'objet résultant d'une construction à l'aide d'un tel opérateur, il est nécessaire de respecter certaines contraintes. Celles-ci ne peuvent pas être systématiquement vérifiées de manière algorithmique, ce qui nécessite une certaine rigueur pour l'utilisation de ces opérateurs.

Les objets ainsi construits sont assimilés à des objets primitifs, en ce sens que des algorithmes spécifiques sont mis en place pour chacun des opérateurs constructifs considérés.

2.2.3 Test d'intériorité

Le test d'intériorité est essentiel dans le contexte de l'optique. En effet, il permet de connaître la valeur de l'indice de réfraction optique en un point donné. Soit O un objet CSG. Soit x un point de \mathbb{R}^3 . Le test d'intériorité consiste à déterminer la position de x par rapport à O , c'est-à-dire à savoir si $x \in \text{int}(O)$ ou $x \in \text{ext}(O)$, où, par convention, on a noté $\text{int}(O)$ l'intérieur de O et $\text{ext}(O)$ l'extérieur de O . On admet la relation :

$$\text{int}(O) \cup \text{ext}(O) \cup \text{cont}(O) = X ,$$

où $\text{cont}(O)$ est le contour de l'objet O .

Cette notion de position d'un point par rapport à un objet n'a pas de sens pour des objets linéiques ou surfaciques. Le test d'intériorité n'a donc de sens que pour les objets volumiques.

Trois cas se présentent. Si O est un objet primitif P , alors le test d'intériorité est propre à P . Ce test n'a lieu d'être que sur des objets primitifs volumiques. Si P est décrit par une fonction implicite f , alors le test d'intériorité s'écrit, par convention sur le signe de f :

$$x \in \text{int}(P) \iff f(x) < 0 .$$

Si $O = (G, O_1)$ où G est un opérateur géométrique alors le test d'intériorité revient à une transformation géométrique sur x et un test d'intériorité sur O_1 , autrement dit :

$$x \in \text{int}(G, O_1) \iff G^{-1}(x) \in \text{int}(O_1) .$$

Si $O = (E, O_1, \dots, O_n)$ où E est un opérateur ensembliste alors le test d'intériorité se résume à l'évaluation d'une expression booléenne :

$$x \in \text{int}(O) \iff (E, x \in \text{int}(O_1), \dots, x \in \text{int}(O_n)) .$$

L'algorithme de test d'intériorité se résume donc à une étude de cas (algorithme 1).

Algorithme 1: *Position d'un point par rapport à un objet CSG.*

POSITION(x, O)

- (1) **switch** O
- (2) **case** $O = P$
- (3) **return** POSITIONOBJETPRIMITIF(x, P)
- (4) **case** $O = (G, O_1)$
- (5) **return** POSITION($G^{-1}(x), O_1$)
- (6) **case** $O = (E, O_1, \dots, O_n)$
- (7) **return** EVALUER($E, \text{POSITION}(x, O_1), \dots, \text{POSITION}(x, O_n)$)

L'algorithme POSITIONOBJETPRIMITIF détermine la position d'un point par rapport à un objet primitif. Il renvoie *int* si le point est à l'intérieur de l'objet considéré, *ext* s'il est à l'extérieur. L'algorithme EVALUER évalue une expression booléenne et renvoie *int* ou *ext*.

Un point se situant sur le contour d'un objet CSG constitue un cas ambigu. Soit, par convention, il suffit de considérer qu'un tel point est à l'intérieur ou à l'extérieur de l'objet. Soit on interdit l'utilisation de l'algorithme pour de tels points. Dans notre contexte, puisque l'intérêt est de connaître l'indice de réfraction optique des milieux, et que cet indice n'a pas de valeur sur les contours des objets, il n'est donc pas nécessaire d'accorder une importance particulière à ce cas ambigu.

2.2.4 Intersection entre un rayon et un objet

Soit O un objet CSG. Soit un rayon caractérisé par un point x et une direction w (appartenant à la sphère unité, c'est-à-dire $\|w\| = 1$), noté $R_{x,w}$. On pose par définition :

$$R_{x,w} = \{y \in \mathbb{R}^3, y = x + tw, t > 0\} .$$

On se propose de trouver, s'il existe, le premier point d'intersection entre le rayon et l'objet O , c'est-à-dire le point x_{\min} défini par :

$$x_{\min} = \xi_O(x, w) ,$$

où ξ_O est la fonction de lancer de rayon⁵ (figure 2.7) :

$$\xi_O(x, w) = x + \nu_O(x, w) w ,$$

où ν_O est la fonction distance aux surfaces⁵ définies par l'objet CSG O (figure 2.7) :

$$\nu_O(x, w) = \begin{cases} \inf \{t > 0, x + tw \in O\} & \text{si } R_{x,w} \cap O \neq \emptyset \\ +\infty & \text{sinon} \end{cases} .$$

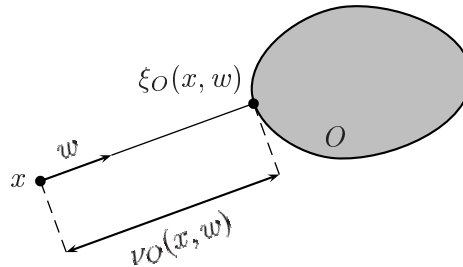


Fig. 2.7 – Fonctions distance ν_O et lancer de rayon ξ_O .

La recherche de la première intersection entre le rayon $R_{x,w}$ et le contour de l'objet O consiste dans un premier temps à calculer l'intersection entre $R_{x,w}$ et tous les objets primitifs (y compris les objets obtenus à partir d'opérateurs constructifs) qui constituent O , en tenant compte des opérateurs géométriques rencontrés. Ensuite il faut trier ces intersections, par rapport aux opérateurs ensemblistes rencontrés, de manière à ne retenir que la plus proche de l'origine du rayon et qui se trouve sur le contour de l'objet O . Nous allons présenter deux méthodes qui se différencient essentiellement par la manière de considérer et trier les intersections :

- la méthode utilisée dans le contexte industriel de l'optique, appelée méthode des epsilon,
- une méthode plus classique, appelée méthode de Roth (Roth [69]).

Méthode des epsilon

Si O est un objet primitif P (y compris obtenu à partir d'opérateurs constructifs), alors le calcul d'intersection est propre à P .

Si $O = (G, O_1)$ où G est un opérateur géométrique alors il est nécessaire de placer le rayon dans le repère de l'objet O_1 , d'effectuer le calcul d'intersection et de replacer l'intersection obtenue dans le repère d'origine, autrement dit :

$$R_{x,w} \cap (G, O_1) = G (G^{-1}(R_{x,w}) \cap O_1) .$$

⁵Voir les définitions données au chapitre 1 page 21.

Si $O = (E, O_1, \dots, O_n)$ où E est un opérateur ensembliste, alors il faut calculer l'intersection avec chaque objet O_i , puis trier ces intersections en fonction de l'opérateur ensembliste E . Les algorithmes de calcul d'intersection entre un rayon et un objet renvoient un couple (x_{\min}, t_{\min}) composé du point d'intersection recherché et du paramètre correspondant sur le rayon :

$$x_{\min} = x + t_{\min} w .$$

La recherche de la première intersection entre $R_{x,w}$ et chaque objet O_i est un couple $(x_{\min,i}, t_{\min,i})$. Le tri des points d'intersection ainsi obtenus consiste à ne retenir que le point le plus proche de l'origine du rayon, c'est-à-dire le point correspondant au paramètre le plus petit (figure 2.8) :

$$t_{\min} = \min_i (t_{\min,i}) .$$

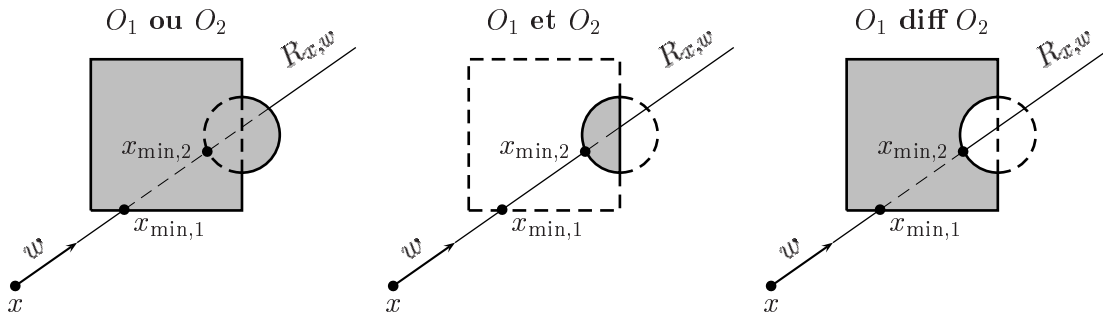


Fig. 2.8 – Méthode des epsilons.

Ensuite, il reste à vérifier que x_{\min} est bien sur le contour de l'objet O . Pour ce faire, il est nécessaire d'éliminer les surfaces virtuelles, c'est-à-dire qui ne sont pas sur le contour de l'objet O (pointillés sur la figure 2.8), ce que permet le test d'intériorité. Il suffit de vérifier que le rayon traverse le contour de O au point x_{\min} , autrement dit que la propriété suivante est vérifiée (figure 2.9) :

$$\exists \epsilon \text{ tel que } \begin{cases} x \in \text{int}(O) \implies (x_{\min} + \epsilon w) \in \text{ext}(O) \\ x \in \text{ext}(O) \implies (x_{\min} + \epsilon w) \in \text{int}(O) \end{cases} .$$

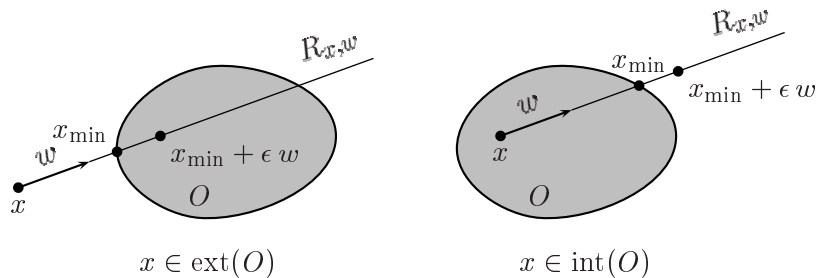


Fig. 2.9 – Le rayon traverse le contour de l'objet.

L'algorithme de recherche de la première intersection entre $R_{x,w}$ et O consiste en une étude de cas et un parcours récursif de l'arbre syntaxique représentant O (algorithme 2).

Algorithme 2: *Méthode des epsilons.*

```

INTERRAYONOBJETCSG-ME( $R_{x,w}, O$ )
(1)  switch  $O$ 
(2)  case  $O = P$ 
(3)    return INTERRAYONOBJETPRIMITIF( $R_{x,w}, P$ )
(4)  case  $O = (G, O_1)$ 
(5)     $(x', w') \leftarrow (G^{-1}(x), G^{-1}(w))$ 
(6)     $(t'_{\min}, I_{\min}) \leftarrow$  INTERRAYONOBJETCSG-ME( $R_{x',w'}, O_1$ )
(7)    if  $t'_{\min} < +\infty$ 
(8)       $t_{\min} \leftarrow \|G(x' + t'_{\min} w') - x\|$ 
(9)    else
(10)      $t_{\min} \leftarrow +\infty$ 
(11)    return  $(t_{\min}, I_{\min})$ 
(12)  case  $O = (E, O_1, \dots, O_n)$ 
(13)     $trouve \leftarrow$  false
(14)     $posini \leftarrow$  POSITION( $x, O$ )
(15)    while not  $trouve$ 
(16)       $(t_{\min}, I_{\min}) \leftarrow (+\infty, \{\})$ 
(17)      for  $i = 1$  to  $n$ 
(18)         $(t, I) \leftarrow$  INTERRAYONOBJETCSG-ME( $R_{x,w}, O_i$ )
(19)        if  $t < t_{\min}$ 
(20)           $(t_{\min}, I_{\min}) \leftarrow (t, I)$ 
(21)        if  $t_{\min} = +\infty$ 
(22)           $trouve \leftarrow$  true
(23)        else
(24)           $x_{\min} \leftarrow x + t_{\min} w$ 
(25)          choisir  $\epsilon$  petit
(26)           $pos \leftarrow$  POSITION( $x_{\min} + \epsilon w, O$ )
(27)          if  $pos = posini$ 
(28)             $x \leftarrow x_{\min} + \epsilon w$ 
(29)          else
(30)             $trouve \leftarrow$  true
(31)    return  $(t_{\min}, I_{\min})$ 

```

L'algorithme INTERRAYONOBJETPRIMITIF calcule la première intersection entre un rayon et un objet primitif. Si une intersection existe, alors il renvoie le paramètre correspondant sur le rayon et, éventuellement, diverses informations relatives à l'objet et au point d'intersection trouvé (une référence à l'objet primitif considéré, le point dans l'espace des paramètres pour un objet paramétrique, etc). Sinon, par convention, il renvoie $(+\infty, \{\})$. Le choix d'une valeur ϵ pour se déplacer le long du rayon, en vue de vérifier si le rayon traverse le contour d'un objet volumique, n'est pas trivial. Il doit être suffisamment petit pour ne pas traverser à nouveau le contour de l'objet considéré, mais pas trop petit pour des raisons numériques (précision, tolérances géométriques). Son choix est d'autant plus difficile dans le cas de rayons proches du plan tangent à l'objet considéré ou proches de parties de l'objet à forte courbure.

Méthode de Roth

La méthode de Roth (Roth [69]) se différencie de la méthode des epsilons au niveau du calcul des points d'intersection. En effet, les algorithmes de calcul d'intersection entre un rayon et un objet renvoient une liste d'intervalles d'intersection L qui correspond aux intersections avec le contour de

l'objet considéré (figure 2.10). C'est une liste de doublets de nombres réels correspondants à des valeurs du paramètre sur le rayon. Elle est ordonnée de manière croissante :

$$\begin{cases} L = \{(t_{1,1}, t_{1,2}), \dots, (t_{m,1}, t_{m,2})\} \\ t_{1,1} \leq t_{1,2} < \dots < t_{m,1} \leq t_{m,2} \end{cases} .$$

Pour les objets surfaciques on a $t_{i,1} = t_{i,2}$ puisque ces objets n'ont pas d'épaisseur.

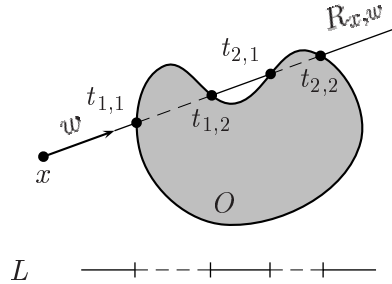


Fig. 2.10 – Liste d'intervalles d'intersection.

Si O est un objet primitif P (y compris obtenu à partir d'opérateurs constructifs), alors le calcul d'intersection est propre à P .

Si $O = (G, O_1)$ où G est un opérateur géométrique alors on est ramené au même cas que dans la méthode des epsilons, c'est-à-dire :

$$R_{x,w} \cap (G, O_1) = G (G^{-1}(R_{x,w}) \cap O_1) .$$

Si $O = (E, O_1, \dots, O_n)$ où E est un opérateur ensembliste, alors il faut calculer la liste d'intervalles d'intersection avec chaque O_i , puis trier ces listes en fonction de l'opérateur ensembliste E . L'intersection entre $R_{x,w}$ et chaque objet O_i est caractérisé par une liste d'intervalles d'intersection L_i . La liste d'intervalles d'intersection entre $R_{x,w}$ et O résulte de l'application de l'opérateur E aux listes L_1, \dots, L_n (figure 2.11). Le point d'intersection recherché correspond à la première valeur, notée $t_{1,1}$, dans la liste L résultante :

$$x_{\min} = x + t_{1,1} w .$$

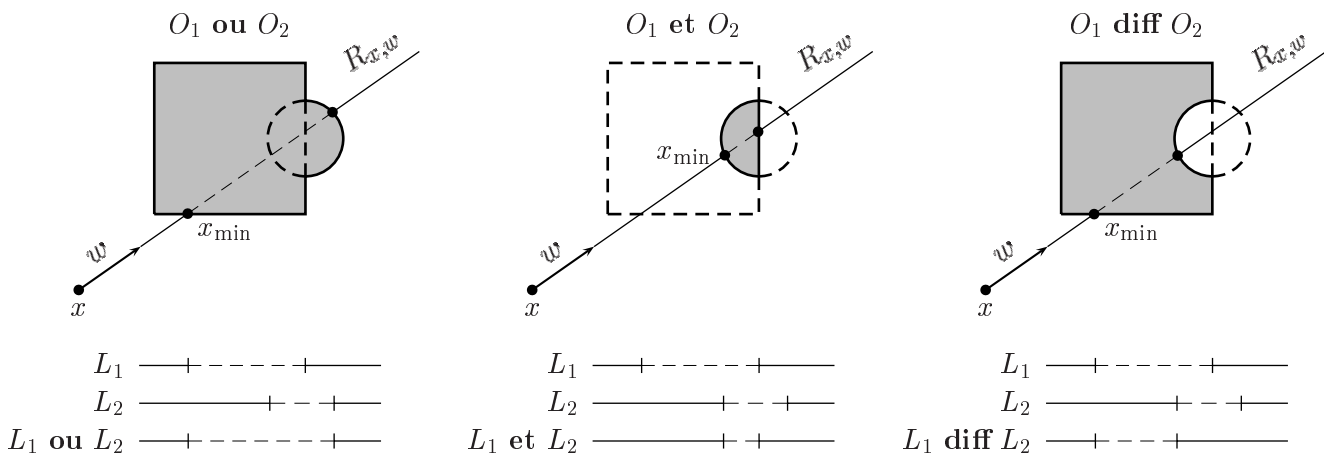


Fig. 2.11 – Méthode de Roth.

La première intersection entre $R_{x,w}$ et O est donc donnée par le premier élément de la liste d'intervalles d'intersection associée à O (algorithme 3).

Algorithme 3: *Méthode de Roth.*

INTERRAYONOBJETCSG-MR($R_{x,w}, O$)

- (1) $(L, I) \leftarrow \text{LISTEINTERRAYONOBJETCSG}(R_{x,w}, O)$
- (2) **return** $(L[1], I)$

L'algorithme LISTEINTERRAYONOBJETCSG calcule la liste d'intervalles d'intersection pour un rayon et un objet donnés (algorithme 4). Si une intersection existe, alors il renvoie une liste d'intervalles d'intersection et, éventuellement, des informations relatives à l'objet et au point d'intersection trouvé (une référence aux objets primitifs considérés, les points dans l'espace des paramètres pour des objets paramétriques, etc). Sinon, par convention, il renvoie $(\emptyset, \{\})$.

Algorithme 4: *Méthode de Roth (liste d'intervalles d'intersection).*

LISTEINTERRAYONOBJETCSG($R_{x,w}, O$)

- (1) **switch** O
- (2) **case** $O = P$
- (3) **return** LISTEINTERRAYONOBJETPRIMITIF($R_{x,w}, P$)
- (4) **case** $O = (G, O_1)$
- (5) $(x', w') \leftarrow (G^{-1}(x'), G^{-1}(w'))$
- (6) $(L', I) \leftarrow \text{LISTEINTERRAYONOBJETCSG}(R_{x',w'}, O_1)$
- (7) **return** $(G(L'), I)$
- (8) **case** $O = (E, O_1, \dots, O_n)$
- (9) **for** $i = 1$ **to** n
- (10) $(L_i, I_i) \leftarrow \text{LISTEINTERRAYONOBJETCSG}(R_{x,w}, O_i)$
- (11) **return** LISTEEVALUER($E, L_1, \dots, L_n, I_1, \dots, I_n$)

L'algorithme LISTEINTERRAYONOBJETPRIMITIF renvoie la liste d'intervalles d'intersection d'un rayon avec un objet primitif et, éventuellement, diverses informations. L'algorithme LISTEEVALUER évalue une liste d'intervalles d'intersection à partir d'un opérateur ensembliste appliqué à d'autres listes et, éventuellement, met à jour les informations relatives aux objets et aux intersections.

2.2.5 Équivalence avec un arbre binaire

Commençons par rappeler quelques définitions et notions élémentaires relatives aux arbres⁶. Un arbre est un graphe orienté particulier, c'est-à-dire une ensemble de nœuds dont certains sont reliés par des branches orientées. Si une branche mène d'un nœud a vers un nœud b alors on dira que a est le père de b ou bien que b est le fils de a (figure 2.12). Puisque l'orientation des branches dans un arbre est intuitive on se contentera de les représenter par un trait (sans flèche).

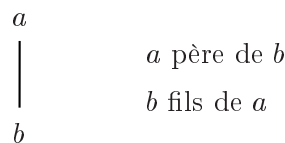


Fig. 2.12 – Définition des nœuds père et fils.

⁶Pour plus de précision, voir des ouvrages généraux tel que celui écrit par Froidevaux, Gaudel et Soria [40].

Les nœuds qui n'ont pas de fils sont appelés feuilles, les autres sont appelés nœuds internes. Un seul nœud n'a pas de père, c'est la racine de l'arbre. Le niveau d'un nœud est égal à un plus la longueur du plus court chemin pour aller de la racine de l'arbre à ce nœud. Par définition, la racine est un nœud de niveau 1. La hauteur d'un arbre est le niveau maximal.

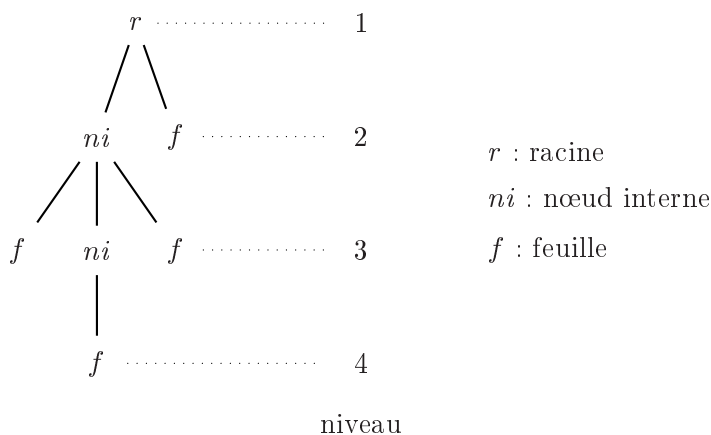


Fig. 2.13 – Notations pour un arbre.

Un arbre binaire est un arbre dont chaque nœud interne possède exactement deux fils. Le dénombrement des nœuds dans de tels arbres est trivial. En effet, un arbre binaire comportant m feuilles contient $m - 1$ nœuds internes, donc au total $2m - 1$ nœuds. Deux formes d'arbre binaire sont particulièrement intéressantes : les arbres binaires équilibrés et les arbres binaires dégénérés (figure 2.14). Les hauteurs de ces arbres sont les suivantes :

$$h = \begin{cases} 1 + \log_2(m) & \text{si arbre équilibré} \\ m & \text{si arbre dégénéré} \end{cases} ,$$

où m est le nombre de feuilles de l'arbre considéré.

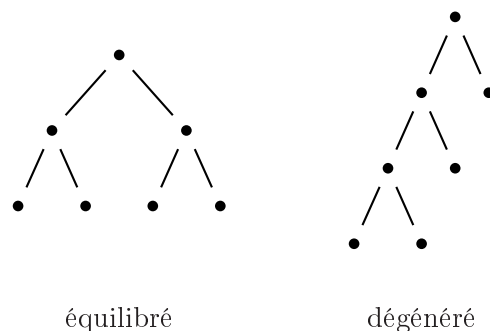


Fig. 2.14 – Arbres binaires équilibré et dégénéré.

Un objet CSG est représenté par un arbre quelconque. L'intérêt de la proposition suivante est de donner une représentation équivalente par un arbre binaire. Cela permettra, entre autres, d'utiliser les expressions simples de dénombrements des nœuds internes et des feuilles. De plus, la représentation des objets CSG sera ainsi fixée pour la suite.

Proposition 1

Tout objet CSG, issu du modèle géométrique considéré, peut être représenté par un arbre binaire, dont tous les éléments contiennent un opérateur géométrique, éventuellement l'identité.

Donc, un nœud interne de cet arbre contient un opérateur géométrique et un opérateur ensembliste, et une feuille contient un opérateur géométrique et un objet primitif (y compris les objets construits à partir des opérateurs constructifs).

□

Preuve Le principe de la démonstration est d'étudier trois cas de figure envisageables pour un objet CSG (objet primitif, objet construit à partir d'un opérateur géométrique et objet construit à partir d'un opérateur ensembliste).

Soit O est un objet primitif (y compris les objets construits à partir des opérateurs constructifs). L'arbre le représentant se réduit à un seul élément, c'est donc bien un arbre binaire. On peut associer à la racine de cet arbre l'opérateur géométrique identité.

Soit $O = (G, O_1)$ où G est un opérateur géométrique. Considérons que la proposition est vérifiée pour le sous-arbre représentant l'objet O_1 . L'objet O est représenté par un arbre unaire. On peut simplifier cet arbre en plaçant l'opérateur géométrique au niveau de l'objet sur lequel il opère (figure 2.15). On a donc remplacé une branche unaire de l'arbre par un seul élément c'est donc bien un arbre binaire. Puisque l'arbre représentant O_1 est supposé vérifier la proposition, l'arbre représentant O vérifie aussi la proposition.



Fig. 2.15 – Convention de représentation d'un opérateur géométrique.

Soit $O = (E, O_1, \dots, O_n)$ où E est un opérateur ensembliste. Considérons que la proposition est vérifiée pour les sous-arbres représentant les objets O_1, \dots, O_n . Si $E = \mathbf{diff}$, alors la proposition est bien vérifiée puisque c'est un opérateur binaire, et que l'on peut lui associer l'opérateur géométrique identité. Si $E = \mathbf{ou}$ ou $E = \mathbf{et}$, alors il suffit de remarquer que l'arbre syntaxique représentant ce type d'opérateur est équivalent à un arbre binaire (figure 2.16). On peut lui associer l'opérateur géométrique identité, la proposition est donc bien vérifiée.

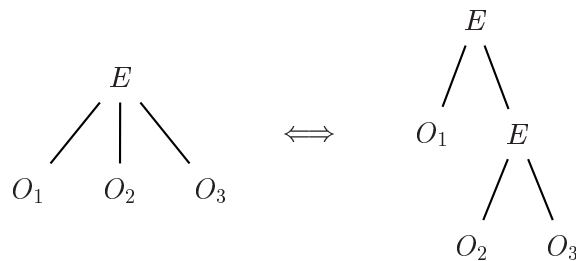


Fig. 2.16 – Hypothèse de représentation pour les opérateurs ensemblistes **ou** et **et**.

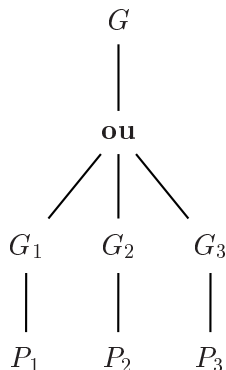
■

Pour appliquer la proposition 1, on procède en deux étapes. Tout d'abord, on regroupe les branches unaires issues d'un opérateur géométrique. On obtient ainsi un arbre dont les nœuds contiennent deux éléments : un opérateur géométrique et un opérateur ensembliste ou un objet primitif. Ensuite, on

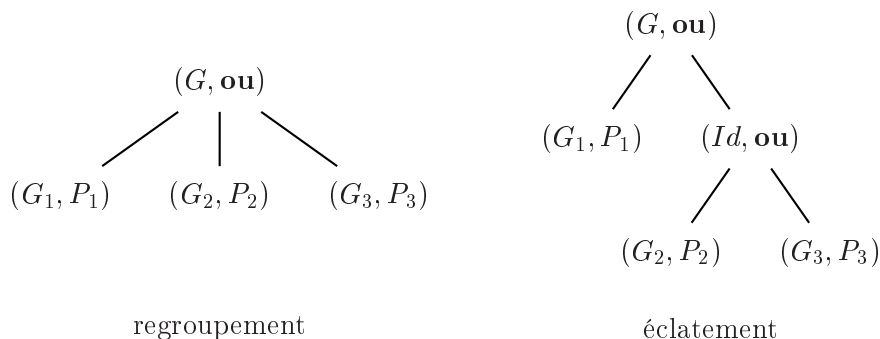
éclate les nœuds comportant au moins trois fils, en prenant garde de ne transmettre que l'opérateur ensembliste (il faut associer aux nouveaux nœuds ainsi créés l'opérateur géométrique identité).

Exemple 1

Soit un objet O construit par union de trois objets primitifs P_1 , P_2 et P_3 . Des opérateurs géométriques G_1 , G_2 et G_3 sont appliqués respectivement à P_1 , P_2 et P_3 . Sur l'objet résultant un opérateur géométrique G est appliqué. L'arbre CSG associé à O comporte 8 nœuds.



La proposition 1 précise qu'une représentation par un arbre binaire équivalente à cet arbre existe. Pour la trouver on commence par regrouper les branches unaires issues d'opérateurs géométriques, puis on éclate les nœuds comportant plus de deux fils.



□

2.2.6 Complexité en mémoire

Dans ce paragraphe, on cherche à évaluer la complexité en mémoire d'une scène géométrique de type CSG, c'est-à-dire d'une liste d'objets CSG O^1, \dots, O^N . Cette complexité représente la place pour stocker toutes les données définissant cette liste d'objets, on la note M_{O^1, \dots, O^N} , et on a la relation suivante :

$$M_{O^1, \dots, O^N} = \sum_{k=1}^N M_{O^k}, \quad (2.2)$$

où M_{O^k} est la complexité en mémoire pour l'objet O^k . Pour calculer cette complexité en mémoire, il est donc nécessaire d'évaluer la complexité en mémoire pour un objet CSG.

Proposition 2

Notons M_P la complexité en mémoire pour l'objet primitif P et $M_{\mathcal{O}_p}$ la complexité en mémoire pour l'opérateur \mathcal{O}_p . Alors la complexité en mémoire pour un objet CSG O s'écrit de manière récursive :

$$M_O = \begin{cases} M_P & \text{si } O = P \\ M_{\mathcal{O}_p} + \sum_{i=1}^n M_{O_i} + n & \text{si } O = (\mathcal{O}_p, O_1, \dots, O_n) \end{cases} .$$

□

Preuve Il faut distinguer deux cas. Si O est un objet primitif P (y compris les objets construits à partir d'opérateurs constructifs) alors la complexité en mémoire est propre à P , et on a :

$$M_O = M_P .$$

Si $O = (\mathcal{O}_p, O_1, \dots, O_n)$ alors il faut compter le nombre de paramètres caractérisant l'opérateur \mathcal{O}_p , rajouter les complexités en mémoire pour les objets O_i et tenir compte des liens vers les objets. Chaque lien peut être vu comme un pointeur vers un objet, la complexité en mémoire associée est 1. Ainsi, la complexité en mémoire pour un tel objet s'écrit de manière récursive :

$$M_O = M_{\mathcal{O}_p} + \sum_{i=1}^n M_{O_i} + n .$$

■

Cette expression récursive n'est pas très explicite, c'est pourquoi on propose de donner une expression de la complexité en mémoire d'un objet CSG qui soit fonction des objets primitifs considérés, et des opérateurs rencontrés. Pour ce faire on utilise la proposition 1 qui permet de travailler avec un arbre binaire.

Proposition 3

Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m . Notons M_P la complexité moyenne en mémoire sur l'ensemble des objets primitifs considérés :

$$M_P = \frac{1}{m} \sum_{i=1}^m M_{P_i} .$$

Notons M_G la complexité moyenne en mémoire sur l'ensemble des opérateurs géométriques associés aux éléments de l'arbre binaire :

$$M_G = \frac{1}{2^m - 1} \sum_{j=1}^{2^m - 1} M_{G,j} ,$$

où $M_{G,j}$ est la complexité en mémoire de l'opérateur géométrique associé à l'élément j de l'arbre. La complexité en mémoire s'écrit de la manière suivante :

$$M_O = (2 M_G + M_P + 3) m - (M_G + 3) . \quad (2.3)$$

□

Preuve La proposition 1 précise que l'objet CSG O peut être représenté par un arbre binaire. Pour comptabiliser la complexité en mémoire de cet arbre, il suffit de recenser les différentes complexités relatives aux nœuds. Un nœud interne contient un opérateur géométrique et opérateur ensembliste,

dont la complexité en mémoire vaut 1 (un entier suffit à caractériser un tel opérateur). Un arbre binaire contient $m - 1$ nœuds internes, la complexité en mémoire associée à l'ensemble des nœuds internes s'écrit donc :

$$\sum_{j=1}^{m-1} (M_{G,j} + 1) .$$

Une feuille contient un opérateur géométrique et un objet primitif. Un arbre binaire contient m feuilles, la complexité en mémoire associée à l'ensemble des feuilles s'écrit donc :

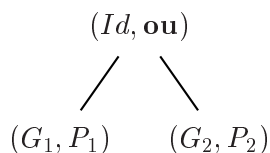
$$\sum_{i=1}^m (M_{G,i} + M_{P_i}) .$$

Pour trouver l'expression proposée, il suffit de sommer les deux quantités précédentes et de rajouter la complexité en mémoire des liens vers les objets, à savoir $2(m - 1)$, puis d'injecter les expressions de M_P et de M_G . ■

Pour se donner un ordre de grandeur, remarquons qu'un opérateur ensembliste est caractérisé par un seul paramètre et que les opérateurs géométriques de base sont caractérisés par quelques paramètres. La translation nécessite un vecteur, la rotation un point, un vecteur et un angle et l'homothétie un point et trois nombres réels.

Exemple 2

Soit un objet O construit par union d'une sphère P_1 et d'un cylindre P_2 . Le nombre d'objets primitifs est donc $m = 2$. Un opérateur de translation G_1 est appliqué à la sphère tandis qu'un opérateur de rotation G_2 est appliqué au cylindre. Aucun opérateur géométrique n'est appliqué sur l'objet résultant de l'union.



La sphère est caractérisée par un rayon, donc $M_{P_1} = 1$, le cylindre est caractérisé par un rayon et une longueur, donc $M_{P_2} = 2$. La complexité moyenne en mémoire sur l'ensemble des objets primitifs est donc $M_P = (1 + 2)/2 = 3/2$.

L'opérateur géométrique de translation G_1 est caractérisé par un vecteur, c'est-à-dire par trois nombres réels, donc $M_{G_1} = 3$. L'opérateur géométrique de rotation est caractérisé par trois angles, donc $M_{G_2} = 3$. L'opérateur géométrique identité est associé à l'objet résultant. La complexité moyenne en mémoire sur l'ensemble des opérateurs géométriques est donc $M_G = (0 + 3 + 3)/3 = 2$.

La complexité en mémoire pour l'objet O s'écrit donc :

$$M_O = (4 + 3/2 + 3) 2 - (2 + 3) = 12 .$$

□

2.2.7 Complexité en temps

Dans ce paragraphe, on cherche à évaluer la complexité en temps de recherche de la première intersection entre un rayon $R_{x,w}$ et une scène géométrique de type CSG, c'est-à-dire une liste d'objets CSG O^1, \dots, O^N , pour une méthode naïve (algorithme 6 page 90). Cette complexité représente le coût

en temps de l'exécution de l'algorithme qui retourne, s'il existe, le point d'intersection recherché, on la note $T_{O^1, \dots, O^N}(x, w)$, et on a la relation suivante (proposition 13 page 90) :

$$T_{O^1, \dots, O^N}(x, w) = \sum_{k=1}^N T_{O^k}(x, w) , \quad (2.4)$$

où $T_{O^k}(x, w)$ est la complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et l'objet O^k . Pour calculer cette complexité en temps il est nécessaire d'évaluer la complexité en temps pour un objet CSG. Et pour ce faire, il faut distinguer les deux méthodes envisagées. Dans chaque méthode, on se donnera un objet CSG et on différenciera les trois cas suivants :

- O est un objet primitif P (y compris les objets construits à partir d'opérateurs constructifs),
- $O = (G, O_1)$ où G est un opérateur géométrique,
- $O = (E, O_1, \dots, O_n)$ où E est un opérateur ensembliste.

Test d'intériorité

L'étude de la complexité en temps de la méthode des espilons nécessite de connaître la complexité en temps du test d'intériorité (algorithme 1). On considère que la complexité en temps de ce test est indépendante du point considéré. Cette restriction est justifiée si la complexité en temps du test d'intériorité appliqué à un objet primitif est indépendante du point considéré. Tout dépend donc des objets primitifs pris en compte et surtout du test d'intériorité propre à chacun de ces objets. Pour des objets primitifs représentés par une fonction implicite (la sphère par exemple), le test d'intériorité se ramène à une évaluation de fonction. L'hypothèse faite est donc valable. Pour des objets primitifs représentés par un ensemble de fonctions implicites (un cylindre par exemple), le test d'intériorité se ramène à une évaluation de chacune des fonctions. Cependant, une optimisation de la méthode consiste à évaluer chaque fonction tant que le critère d'intériorité est vérifié. Toutes les fonctions ne sont donc pas nécessairement évaluées, la complexité en temps devient donc dépendante du point considéré. Ce qui contredit notre hypothèse. Pour pallier ce problème, il suffit de considérer une complexité moyenne en temps sur les objets primitifs.

Rajoutons à ces justifications le fait que, dans notre contexte, les tests d'intériorité sont appliqués à des points issus d'intersections entre un rayon et un objet, ils sont donc tous situés au voisinage des contours réels de ces objets.

Proposition 4

Notons $T_{\text{pos},P}$ la complexité en temps du test d'intériorité appliqué à un objet primitif P . Notons T_G la complexité en temps relative au calcul de la transformée d'un point par un opérateur géométrique. La complexité en temps du test d'intériorité appliqué à un objet O s'exprime de manière récursive :

$$T_{\text{pos},O} = \begin{cases} T_{\text{pos},P} & \text{si } O = P \\ T_G + T_{\text{pos},O_1} & \text{si } O = (G, O_1) \\ \sum_{i=1}^n T_{\text{pos},O_i} & \text{si } O = (E, O_1, \dots, O_n) \end{cases} .$$

□

Preuve Soit O est un objet primitif (y compris les objets construits à partir d'opérateurs constructifs). Alors la complexité en temps du test d'intériorité est propre à l'objet primitif :

$$T_{\text{pos},O} = T_{\text{pos},P} .$$

Soit $O = (G, O_1)$ où G un opérateur géométrique. La complexité en temps associée consiste en une transformation géométrique sur un point et un test d'intériorité sur l'objet O_1 , d'où une expression récursive :

$$T_{\text{pos},O} = T_G + T_{\text{pos},O_1} .$$

Soit $O = (E, O_1, \dots, O_n)$ où E un opérateur ensembliste. Il suffit de supposer que la complexité en temps de l'évaluation de l'expression booléenne (algorithme EVALUER) est négligeable par rapport aux complexités en temps du test d'intériorité appliqué aux objets O_1, \dots, O_n . La complexité en temps associée s'écrit donc de manière récursive :

$$T_{\text{pos},O} = \sum_{i=1}^n T_{\text{pos},O_i} .$$

■

Utilisons la proposition 1 qui permet de travailler avec un arbre binaire pour établir une expression plus explicite de la complexité en temps.

Proposition 5

Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m . Notons $T_{\text{pos},P}$ la complexité en temps moyennée sur l'ensemble des objets primitifs :

$$T_{\text{pos},P} = \frac{1}{m} \sum_{i=1}^m T_{\text{pos},P_i} .$$

Notons T_G la complexité en temps relative au calcul de la transformée d'un point par un opérateur géométrique. Notons ξ la proportion d'éléments de l'arbre qui sont associés à un opérateur géométrique qui n'est pas l'identité. La complexité en temps du test d'intériorité appliqué à l'objet O s'écrit :

$$T_{\text{pos},O} = (2m - 1)\xi T_G + m T_{\text{pos},P} . \quad (2.5)$$

□

Preuve Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m (y compris les objets construits à partir des opérateurs constructifs). La proposition 1 précise que l'objet CSG O peut être représenté par un binaire. Pour évaluer la complexité en temps du test d'intériorité appliqué à O il suffit de comptabiliser les différentes complexités relatives aux nœuds de cet arbre. Un nœud interne contient un opérateur géométrique et un opérateur ensembliste. L'évaluation booléenne est négligeable devant les autres complexités. Un arbre binaire contient $m - 1$ nœuds internes, la complexité en temps du test d'intériorité associée à l'ensemble des nœuds internes s'écrit donc :

$$\sum_{j=1}^{m-1} T_{G,j} ,$$

où $T_{G,j}$ est la complexité en temps de l'opérateur géométrique associé au nœud j de l'arbre.

Une feuille contient un opérateur géométrique et un objet primitif. Un arbre binaire contient m feuilles, la complexité en temps du test d'intériorité associée à l'ensemble des feuilles s'écrit donc :

$$\sum_{i=1}^m (T_{G,i} + T_{\text{pos},P_i}) .$$

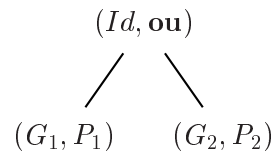
Pour exprimer la complexité en temps sur tous les éléments de l'arbre considéré, il suffit de sommer les deux quantités précédentes puis d'injecter l'expression de $T_{\text{pos},P}$, en remarquant que par définition de ξ on a :

$$\sum_{j=1}^{m-1} T_{G,j} + \sum_{i=1}^m T_{G,i} = (2m - 1) \xi T_G .$$

D'où l'expression proposée pour $T_{\text{pos},O}$. ■

Exemple 3

Soit un objet O construit par union d'une sphère P_1 et d'un cylindre P_2 . Le nombre d'objets primitifs est donc $m = 2$. Un opérateur de translation G_1 est appliqué à la sphère tandis qu'un opérateur de rotation G_2 est appliqué au cylindre. Aucun opérateur géométrique n'est appliqué sur l'objet résultant de l'union.



Il y a 2 opérateurs géométriques dans l'arbre binaire associé à O , leur proportion est donc $\xi = (0 + 1 + 1)/3 = 2/3$. La valeur de la complexité en temps du test d'intériorité appliqué à O est donc :

$$T_{\text{pos},O} = 2T_G + 2T_{\text{pos},P} .$$

□

Méthodes des epsilons

En particulier pour une écriture plus simple des cas où il n'y a pas intersection, on introduit la fonction suivante.

Définition 2

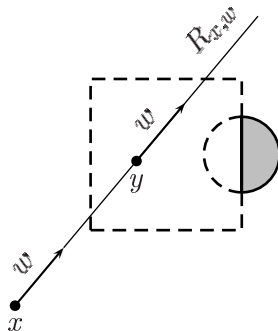
On note $\mathbb{1}_{\mathcal{R},O}$ une fonction définie sur \mathcal{R} à valeurs dans $\{0, 1\}$, dont l'expression est la suivante :

$$\mathbb{1}_{\mathcal{R},O}(x, w) = \begin{cases} 1 & \text{si } R_{x,w} \cap O \neq \emptyset \\ 0 & \text{sinon} \end{cases} . \quad (2.6)$$

La fonction $\mathbb{1}_{\mathcal{R},O}$ est appelée indicatrice d'intersection sur O relativement à \mathcal{R} . □

Pour donner une expression relativement simple de la complexité en temps pour la méthode des epsilons il est nécessaire de faire une hypothèse supplémentaire sur la fonction T_O . On suppose donc que la T_O est constante pour des points se trouvant sur un même rayon et situés dans un voisinage de l'objet O (par rapport aux objets primitifs de O). Soit $R_{x,w}$ un rayon issu de \mathcal{R} . La présente hypothèse signifie que pour des points y se trouvant sur $R_{x,w}$ et situés dans un voisinage de l'objet O alors on a l'égalité suivante :

$$T_O(y, w) = T_O(x, w) .$$

Fig. 2.17 – Hypothèse sur la fonction T_O .

Cette hypothèse est justifiée par le fait que les calculs d'intersection pour des rayons dont les origines se trouvent sur $R_{x,w}$ et de direction w sont semblables puisque ces rayons sont tous portés par le même rayon $R_{x,w}$.

La proposition suivante donne une expression récursive de la complexité en temps pour la méthode des epsilons (algorithme 2).

Proposition 6

Soit $R_{x,w}$ un rayon. Notons $T_P(x, w)$ la complexité en temps de recherche de la première intersection entre le rayon et l'objet primitif P . Notons T_G la complexité en temps relative au calcul de la transformée d'un point ou d'un vecteur par un opérateur géométrique. Notons $\kappa_O(x, w)$ le nombre de surfaces virtuelles rencontrées par le rayon considéré lors de la recherche de la première intersection avec un objet O . La complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et un objet O s'exprime de manière récursive :

$$T_O(x, w) = \begin{cases} T_P(x, w) & \text{si } O = P \\ 3T_G + T_{O_1}(x', w') & \text{si } O = (G, O_1) \\ T_{E,O}(x, w) & \text{si } O = (E, O_1, \dots, O_n) \end{cases},$$

où x' et w' sont les paramètres du rayon $G^{-1}(R_{x,w})$, et $T_{E,O}(x, w)$ est donné par :

$$T_{E,O}(x, w) = (\kappa_O(x, w) + 1) \left(\sum_{i=1}^n (T_{O_i}(x, w) + T_{\text{pos}, O_i}) \right) + \mathbb{1}_{\mathcal{R}, O}(x, w) \left(\sum_{i=1}^n T_{\text{pos}, O_i} \right),$$

où T_{pos, O_i} est la complexité en temps du test d'intériorité appliqué à l'objet O_i .

□

Preuve Soit $R_{x,w}$ un rayon.

Si O est un objet primitif P (y compris les objets construits à partir d'opérateurs constructifs) alors la complexité en temps est propre à l'algorithme de calcul d'intersection du rayon avec P et on a :

$$T_O(x, w) = T_P(x, w).$$

Si $O = (G, O_1)$ où G est un opérateur géométrique, alors il faut décomposer la méthode. Dans un premier temps, le rayon $R_{x,w}$ est transformé à l'aide de l'opérateur G^{-1} :

$$R_{x', w'} = G^{-1}(R_{x, w}).$$

La complexité en temps associée est $2T_G$ (il faut transformer les deux paramètres du rayon). Ensuite, la recherche de la première intersection entre le rayon $R_{x', w'}$ et l'objet O_1 est effectuée. La complexité

en temps associée est $T_{O_1}(x', w')$. Enfin, le point d'intersection recherché, s'il existe, doit être replacé dans le repère de O , ce qui s'obtient par application de l'opérateur géométrique G . La complexité en temps associée est T_G . La complexité en temps totale pour la recherche de la première intersection entre le rayon $R_{x,w}$ et l'objet O s'écrit donc de manière récursive :

$$T_O(x, w) = 3T_G + T_{O_1}(x', w') .$$

Si $O = (E, O_1, \dots, O_n)$ où E est un opérateur ensembliste, alors la complexité en temps associée se ramène à un calcul d'intersection sur les objets O_1, \dots, O_n et à des tests d'intériorité sur l'objet O . Le nombre de ces tests dépend de l'existence d'une intersection entre le rayon $R_{x,w}$ et l'objet O (test conditionnel des lignes 21 à 30 de l'algorithme 2). Ainsi, la complexité en temps recherchée s'écrit de manière récursive :

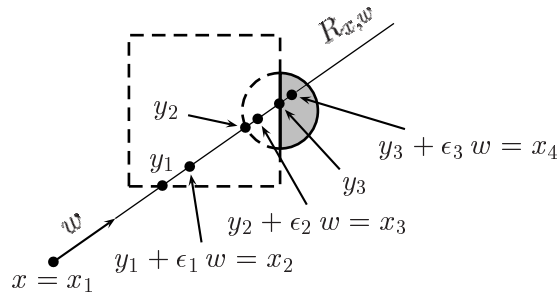
$$T_O(x, w) = T_{\text{pos}, O} + \sum_{j=1}^{\kappa_O(x,w)+1} \left(\sum_{i=1}^n T_{O_i}(x_j, w) + \mathbb{1}_{\mathcal{R}, O_1, \dots, O_n}(x_j, w) T_{\text{pos}, O} \right) ,$$

où $\kappa_O(x, w) + 1$ est le nombre de fois où le calcul de recherche de la première intersection est effectué (boucle conditionnelle des lignes 15 à 30 de l'algorithme 2), ce qui signifie bien que $\kappa_O(x, w)$ est le nombre de surfaces virtuelles rencontrées, $x_1, \dots, x_{\kappa_O(x,w)+1}$ sont les origines successives des différents rayons générés le long du rayon $R_{x,w}$ lors des calculs répétés :

$$\begin{cases} x_j = x_{j-1} + (t_{j-1} + \epsilon_{j-1}) w , & j \in \{2, \dots, \kappa_O(x, w) + 1\} , & \kappa_O(x, w) \geq 1 \\ x_1 = x \end{cases} ,$$

où les t_j sont les distances successives calculées et les ϵ_j sont les petits déplacements effectués le long du rayon pour les tests d'intériorité et, enfin, où $\mathbb{1}_{\mathcal{R}, O_1, \dots, O_n}$ est une fonction définie sur \mathcal{R} à valeurs dans $\{0, 1\}$ qui vaut 1 si au moins une intersection entre $R_{x,w}$ et un des objets O_i existe :

$$\mathbb{1}_{\mathcal{R}, O_1, \dots, O_n}(x_j, w) = \begin{cases} 1 & \text{si } \exists i \in \{1, \dots, n\} \text{ tel que } R_{x_j, w} \cap O_i \neq \emptyset \\ 0 & \text{sinon} \end{cases} .$$



Succession des calculs en se déplaçant le long du rayon.

La succession des calculs pour des rayons dont les origines se trouvent sur le même rayon $R_{x,w}$ et dont la direction est w permet de considérer que la complexité en temps de recherche de la première intersection avec l'objet O_i est constante pour les rayons $R_{x_j, w}$. On a donc les égalités suivantes :

$$T_{O_i}(x_j, w) = T_{O_i}(x, w) , \forall j \in \{1, \dots, \kappa_O(x, w) + 1\} .$$

D'autre part, on remarque que la fonction $\mathbb{1}_{\mathcal{R}, O_1, \dots, O_n}$ vaut nécessairement 1 pour tous les points (x_j, w) tels que $j \in \{1, \dots, \kappa_O(x, w)\}$. En effet, pour ces points il y a bien existence d'une surface

virtuelle ayant une intersection avec le rayon $R_{x_j, w}$. Par conséquent, on peut regrouper les termes contenant $T_{\text{pos}, O}$ de la manière suivante :

$$T_{\text{pos}, O} + \sum_{j=1}^{\kappa_O(x, w)+1} \mathbb{1}_{\mathcal{R}, O_1, \dots, O_n}(x_j, w) T_{\text{pos}, O} = (\kappa_O(x, w) + 1) T_{\text{pos}, O} + \mathbb{1}_{\mathcal{R}, O_1, \dots, O_n}(x_{\kappa_O(x, w)+1}, w) T_{\text{pos}, O} .$$

Dans le dernier terme, le facteur devant $T_{\text{pos}, O}$ correspond à l'indicatrice d'intersection entre le rayon $R_{x, w}$ et l'objet O (c'est le dernier calcul effectué), c'est-à-dire :

$$\mathbb{1}_{\mathcal{R}, O_1, \dots, O_n}(x_{\kappa_O(x, w)+1}, w) = \mathbb{1}_{\mathcal{R}, O}(x, w) .$$

D'après la proposition 4 la complexité en temps du test d'intériorité s'écrit :

$$T_{\text{pos}, O} = \sum_{i=1}^n T_{\text{pos}, O_i} .$$

Ainsi la complexité en temps recherchée s'écrit finalement :

$$T_O(x, w) = (\kappa_O(x, w) + 1) \left(\sum_{i=1}^n (T_{O_i}(x, w) + T_{\text{pos}, O_i}) \right) + \mathbb{1}_{\mathcal{R}, O}(x, w) \left(\sum_{i=1}^n T_{\text{pos}, O_i} \right) .$$

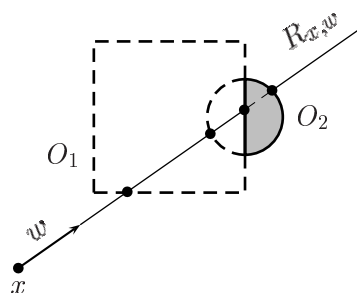
D'où l'expression de la complexité en temps proposée. ■

Remarque

Le nombre de surfaces virtuelles rencontrées $\kappa_O(x, w)$ est relié aux nombres d'intersections $\lambda_{O_i}(x, w)$ entre le rayon $R_{x, w}$ et les objets O_i par l'inégalité suivante :

$$0 \leq \kappa_O(x, w) \leq \sum_{i=1}^n \lambda_{O_i}(x, w) . \quad (2.7)$$

Par contre, il n'y a pas de relation particulière entre $\kappa_O(x, w)$ et $\lambda_O(x, w)$.



$$\begin{aligned} O &= O_2 \text{ diff } O_1 \\ \kappa_O(x, w) &= 2 \\ \lambda_O(x, w) &= 3 \\ \lambda_{O_1}(x, w) + \lambda_{O_2}(x, w) &= 4 \end{aligned}$$

Lien entre le nombre de surfaces virtuelles rencontrées et le nombre d'intersections.

De même que pour le nombre d'intersections, le nombre de surfaces virtuelles rencontrées est une fonction de $L^1(\mathcal{R})$ continue par morceaux. □

La définition suivante donne une caractérisation de la proportion de rayons issus de \mathcal{R} qui ont une intersection avec un objet.

Définition 3

La proportion de rayons issus de \mathcal{R} qui ont une intersection avec un objet O est donnée par le nombre réel suivant :

$$\gamma_{\mathcal{R},O} = \frac{1}{\text{mes}(\mathcal{R})} \int_{\mathcal{R}} \mathbb{1}_O(x, w) d\mu(x) d\sigma(w) . \quad (2.8)$$

□

La définition précédente est cohérente. En effet, l'intégrale au numérateur a bien un sens puisque l'indicatrice d'intersection est une fonction bornée, donc dans l'espace métrique $L^1(\mathcal{R})$. D'autre part, on remarque que $\gamma_{\mathcal{R},O} \in [0, 1]$, et, par définition de l'indicatrice d'intersection, ce nombre représente bien la proportion de rayons issus de \mathcal{R} qui ont une intersection avec l'objet O considéré.

Le corollaire suivant exprime la complexité moyenne en temps pour la méthode des epsilons (algorithme 2).

Corollaire 1

Soit \mathcal{R}_j un sous-ensemble de \mathcal{R} où κ_O est constante. La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R}_j et un objet O s'écrit de la manière suivante :

$$\bar{T}_O = \begin{cases} \bar{T}_P & \text{si } O = P \\ 3T_G + \bar{T}_{O_1} & \text{si } O = (G, O_1) \\ \bar{T}_{E,O} & \text{si } O = (E, O_1, \dots, O_n) \end{cases} ,$$

où $\bar{T}_{E,O}$ est donné par :

$$\bar{T}_{E,O} = (\kappa_O|_{\mathcal{R}_j} + 1) \left(\sum_{i=1}^n (\bar{T}_{O_i} + T_{\text{pos},O_i}) \right) + \gamma_{\mathcal{R}_j,O} \left(\sum_{i=1}^n T_{\text{pos},O_i} \right) ,$$

où $\kappa_O|_{\mathcal{R}_j}$ est la valeur de κ_O sur \mathcal{R}_j .

□

Preuve Il suffit d'appliquer la définition 1 de la complexité moyenne en temps par rapport à l'ensemble de rayons issus de \mathcal{R}_j à l'expression de la complexité de la proposition 6.

■

Remarque

Le corollaire 1 exprime la valeur de la complexité moyenne en temps sur un sous-ensemble de \mathcal{R} . Les cas $O = P$ et $O = (G, O_1)$ peuvent être écrits pour les rayons issus de \mathcal{R} , mais le dernier cas $O = (E, O_1, \dots, O_n)$ n'est pas trivial. En effet, pour ce cas il est nécessaire de savoir écrire la moyenne sur \mathcal{R} de produits de deux fonctions :

$$\overline{\kappa_O T_{O_i}} = \frac{1}{\text{mes}(\mathcal{R})} \int_{\mathcal{R}} \kappa_O(x, w) T_{O_i}(x, w) d\mu(x) d\sigma(w) .$$

L'intégrale au numérateur n'est pas séparable, sauf sur les parties de \mathcal{R} où au moins l'une des deux fonctions est constante. C'est pourquoi on écrit la complexité moyenne en temps sur une partie de \mathcal{R} où κ_O est constante. Cette écriture s'apparente au théorème de la moyenne appliqué à des fonctions continues par morceaux.

□

L'expression récursive du corollaire 1 n'est pas très explicite, c'est pourquoi on propose de donner une expression de la complexité moyenne en temps de la méthode des epsilons qui soit fonction des objets primitifs considérés, et des opérateurs rencontrés. Pour ce faire, on utilise la proposition 1 qui permet de travailler avec un arbre binaire. Commençons par donner un lemme qui caractérise localement dans un arbre CSG la complexité moyenne en temps de recherche de la première intersection. Celui-ci sera utilisé pour la démonstration de la proposition 7.

Lemme 1

Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m . Notons \bar{T}_P la complexité moyenne en temps de recherche de la première intersection moyennée sur l'ensemble des objets primitifs :

$$\bar{T}_P = \frac{1}{m} \sum_{i=1}^m \bar{T}_{P_i} .$$

Notons $T_{\text{pos},P}$ la complexité en temps du test d'intériorité moyennée sur l'ensemble des objets primitifs :

$$T_{\text{pos},P} = \frac{1}{m} \sum_{i=1}^m T_{\text{pos},P_i} .$$

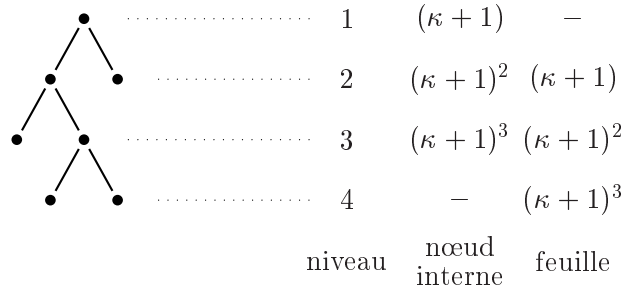
Notons κ le nombre moyen de surfaces virtuelles rencontrées lors de la recherche de la première intersection entre un rayon issu de \mathcal{R} et un sous-objet de O . Notons γ la proportion moyenne de rayons issus de \mathcal{R} et des calculs successifs qui ont une intersection avec un sous-objet de O . Notons T_G la complexité en temps relative au calcul de la transformée d'un point ou d'un vecteur par un opérateur géométrique. Notons ξ la proportion d'éléments de l'arbre qui sont associés à un opérateur géométrique qui n'est pas l'identité. La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et le sous-objet associé à un élément situé au niveau j de l'arbre de O s'écrit :

$$\bar{T}_{O,j} = \begin{cases} (\kappa + 1)^j ((3 + (1 + \gamma)(2m_j - 1)) \xi T_G + (1 + \gamma)m_j T_{\text{pos},P}) & \text{si nœud interne} \\ (\kappa + 1)^{j-1} (3 \xi T_G + \bar{T}_P) & \text{si feuille} \end{cases} ,$$

où m_j est le nombre d'objets primitifs constituant le sous-objet décrit au nœud interne j . □

Preuve Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m (y compris les objets construits à partir d'opérateurs constructifs). La proposition 1 précise que l'objet CSG O peut être représenté par un arbre binaire. Ainsi, chaque nœud interne de l'arbre s'écrit (G, E) et chaque feuille de l'arbre s'écrit (G, P) où G est un opérateur géométrique, E un opérateur ensembliste et P un objet primitif.

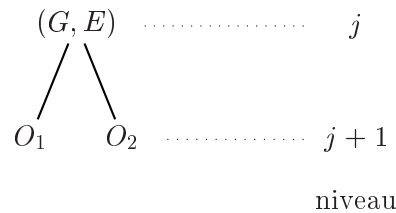
Dire que le nombre moyen de surfaces virtuelles rencontrées est κ c'est dire qu'est associé à chaque nœud de l'arbre un coefficient $\kappa + 1$ rendant compte du nombre de fois où ce nœud, et par conséquent chacun de ses fils, doit être parcouru (élimination des intersections avec des surfaces virtuelles). En effet, il y a au moins un parcours du nœud considéré et un parcours pour chaque surface virtuelle rencontrée, d'où le coefficient $\kappa + 1$.



Nombre de passages sur un nœud interne et sur une feuille lors d'un parcours récursif répété $\kappa + 1$ fois à partir de chaque nœud.

Dire que ξ est la proportion d'éléments de l'arbre qui sont associés à un opérateur géométrique qui n'est pas l'identité, c'est dire que la complexité en temps associée à chaque élément de l'arbre pour les opérateurs géométriques s'écrit $3\xi T_G$. Le facteur 3 provient du nombre de transformations géométriques à effectuer, il y en a deux pour transformer un rayon et une pour transformer un point (une justification précise est donnée dans la démonstration de la proposition 6). Par définition de ξ , ξT_G est la complexité en temps moyennée sur les éléments de l'arbre (une justification précise est donnée dans la démonstration de la proposition 5).

Un nœud interne situé au niveau j dans l'arbre contient un opérateur géométrique et un opérateur ensembliste. Il est nécessaire d'effectuer le test d'intériorité $1 + \gamma$ fois en moyenne (un test d'initialisation et un test supplémentaire s'il y a intersection) au niveau d'un nœud interne, mais il faut répéter le test autant de fois que ce nœud est rencontré lors du parcours de l'arbre, autrement dit $(\kappa + 1)^j$ fois.



Cas d'un nœud interne.

La complexité moyenne en temps associée à un tel nœud interne est donc :

$$\bar{T}_{O,j} = (\kappa + 1)^j (3\xi T_G + (1 + \gamma)T_{\text{pos},j}) ,$$

où $T_{\text{pos},j}$ est la complexité en temps du test d'intériorité associée à un nœud interne situé au niveau j . Cette complexité est donnée par la proposition 5 :

$$T_{\text{pos},j} = (2m_j - 1)\xi T_G + m_j T_{\text{pos},P} ,$$

où m_j est le nombre d'objets primitifs constituant le sous-objet décrit au nœud interne considéré. Donc, la complexité moyenne en temps associée à un nœud interne situé au niveau j s'écrit :

$$\bar{T}_{O,j} = (\kappa + 1)^j ((3 + (1 + \gamma)(2m_j - 1))\xi T_G + (1 + \gamma)m_j T_{\text{pos},P}) .$$

Une feuille située au niveau j de l'arbre CSG contient un opérateur géométrique et un objet primitif. Le calcul d'intersection entre un rayon et cet objet est effectué autant de fois que cette feuille est rencontrée lors du parcours de l'arbre, c'est-à-dire $(\kappa + 1)^{j-1}$ fois.

$$(G, P) \dots\dots\dots j$$

niveau

Cas d'une feuille.

La complexité moyenne en temps associée à un telle feuille est donc :

$$\overline{T}_{O,j} = (\kappa + 1)^{j-1} (3 \xi T_G + \overline{T}_P) .$$

■

Remarque

Le nombre moyen de surfaces virtuelles rencontrées κ est moyenné sur l'ensemble \mathcal{R} , et devrait en toute rigueur être noté $\overline{\kappa}$. C'est par souci de clarté dans les écritures que l'on a adopté cette notation. Même remarque pour la proportion moyenne de rayons γ .

□

La proposition suivante donne une expression de la complexité moyenne en temps pour un objet CSG dans le cas d'arbres binaires particuliers.

Proposition 7

Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m . Considérons les mêmes notations qu'au lemme 1. La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et l'objet O s'écrit :

$$\overline{T}_O = f_G(m, \kappa, \gamma) \xi T_G + f_{\text{pos},P}(m, \kappa, \gamma) T_{\text{pos},P} + f_P(m, \kappa, \gamma) \overline{T}_P . \quad (2.9)$$

Notons $\kappa_0 = \kappa + 1$. Si l'arbre binaire représentant l'objet O est équilibré et que le nombre d'objets primitifs est une puissance de 2 alors les fonctions f_G , $f_{\text{pos},P}$ et f_P sont définies par :

$$\left\{ \begin{array}{l} f_G(m, \kappa, \gamma) = 2m(1+\gamma) \frac{\kappa_0^{1+\log_2(m)} - \kappa_0}{\kappa_0 - 1} + (2-\gamma) \frac{m\kappa_0^{1+\log_2(m)} - \kappa_0}{2\kappa_0 - 1} + 3m\kappa_0^{m-1} \\ f_{\text{pos},P}(m, \kappa, \gamma) = m(1+\gamma) \frac{\kappa_0^{1+\log_2(m)} - \kappa_0}{\kappa_0 - 1} \\ f_P(m, \kappa, \gamma) = m\kappa_0^{m-1} \end{array} \right. .$$

Si l'arbre binaire représentant l'objet O est dégénéré alors les fonctions f_G , $f_{\text{pos},P}$ et f_P sont définies par :

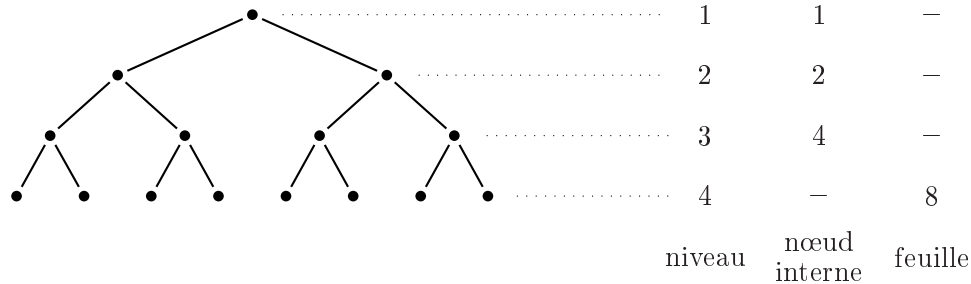
$$\left\{ \begin{array}{l} f_G(m, \kappa, \gamma) = \frac{1}{(\kappa_0 - 1)^2} ((12 + 3\gamma) \kappa_0^{m+1} - (13 + \gamma) \kappa_0^m + 3\kappa_0^{m-1} \\ \quad - (6 + (2m - 1)(1 + \gamma)) \kappa_0^2 + (6 + (2m + 1)(1 + \gamma)) \kappa_0) \\ f_{\text{pos},P}(m, \kappa, \gamma) = (1 + \gamma) \frac{2\kappa_0^{m+1} - \kappa_0^m - (m + 1) \kappa_0^2 + m\kappa_0}{(\kappa_0 - 1)^2} \\ f_P(m, \kappa, \gamma) = \frac{2\kappa_0^m - \kappa_0^{m-1} - \kappa_0}{\kappa_0 - 1} \end{array} \right. .$$

□

Preuve Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m (y compris les objets construits à partir d'opérateurs constructifs). La proposition 1 précise que l'objet CSG O peut être représenté par un arbre binaire.

Pour évaluer la complexité moyenne en temps associée à tous les éléments de l'arbre, ajoutons les complexités moyennes en temps aux nœuds de l'arbre. Pour un arbre binaire équilibré et complet (le nombre de feuilles est une puissance de 2), la complexité moyenne en temps s'écrit en sommant les complexités moyennes en temps associées à chaque niveau :

$$\bar{T}_O = \sum_{j=1}^{h-1} 2^{j-1} \bar{T}_{O,j} + m \bar{T}_{O,m} . \quad (2.10)$$



Nombre de nœuds internes et de feuilles dans un arbre binaire équilibré et complet.

D'après le lemme 1, la complexité moyenne en temps sur l'ensemble des nœuds internes s'écrit :

$$\sum_{j=1}^{h-1} 2^{j-1} \bar{T}_{O,j} = \sum_{j=1}^{h-1} 2^{j-1} (\kappa + 1)^j ((3 + (1 + \gamma) (2m_j - 1)) \xi T_G + (1 + \gamma) m_j T_{\text{pos},P}) . \quad (2.11)$$

Toujours d'après le lemme 1, la complexité moyenne en temps sur l'ensemble des feuilles s'écrit :

$$m \bar{T}_{O,m} = m (\kappa + 1)^{m-1} (3 \xi T_G + \bar{T}_P) . \quad (2.12)$$

Le nombre d'objets primitifs constituant le sous-objet associé au nœud interne situé au niveau j est le nombre de feuilles du sous-arbre associé à ce nœud, on a donc :

$$m_j = \frac{m}{2^{j-1}} .$$

En injectant l'expression de m_j dans 2.11 puis en séparant les différents termes on est amené à calculer les deux sommes suivantes, où, pour simplifier les écritures, l'on a remplacé $\kappa + 1$ par κ_0 :

$$\left\{ \begin{array}{l} \sum_{j=1}^{h-1} 2^{j-1} (\kappa + 1)^j = \frac{2^{h-1} \kappa_0^h - \kappa_0}{2 \kappa_0 - 1} \\ \sum_{j=1}^{h-1} (\kappa + 1)^j = \frac{\kappa_0^h - \kappa_0}{\kappa_0 - 1} \end{array} \right. .$$

Il ne reste plus qu'à regrouper les différents termes obtenus à partir de 2.11 et 2.12 et de les injecter dans 2.10, puis de remarquer que la hauteur d'un arbre binaire équilibré s'écrit :

$$h = 1 + \log_2(m) .$$

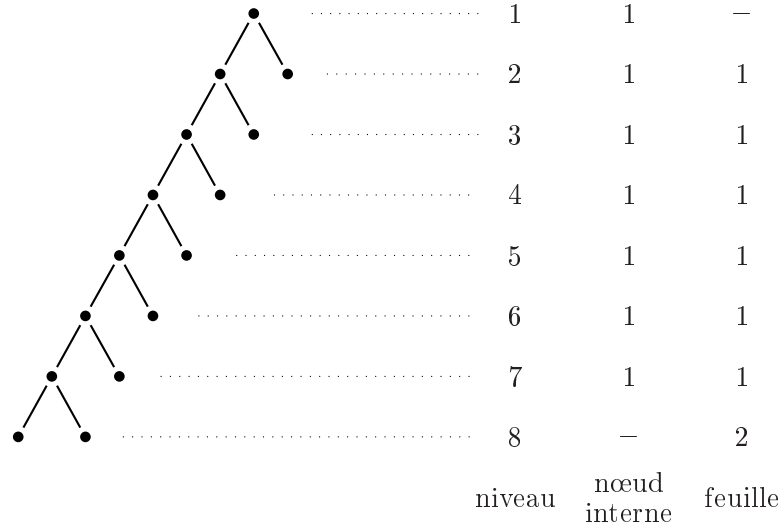
On obtient alors l'expression proposée pour \bar{T}_O :

$$\bar{T}_O = \left(2m(1+\gamma) \frac{\kappa_0^{1+\log_2(m)} - \kappa_0}{\kappa_0 - 1} + (2-\gamma) \frac{m\kappa_0^{1+\log_2(m)} - \kappa_0}{2\kappa_0 - 1} + 3m\kappa_0^{m-1} \right) \xi T_G$$

$$+ m(1+\gamma) \frac{\kappa_0^{1+\log_2(m)} - \kappa_0}{\kappa_0 - 1} T_{\text{pos},P} + m\kappa_0^{m-1} \bar{T}_P .$$

Pour un arbre dégénéré, la complexité en temps s'écrit en sommant les complexités moyenne en temps associées à chaque niveau :

$$\bar{T}_O = \sum_{j=1}^{h-1} \bar{T}_{O,j} + \sum_{j=2}^{h-1} \bar{T}_{O,j} + 2\bar{T}_{O,m} . \quad (2.13)$$



Nombre de nœuds internes et de feuilles dans un arbre binaire dégénéré.

D'après le lemme 1, la complexité moyenne en temps sur l'ensemble des nœuds internes s'écrit :

$$\sum_{j=1}^{h-1} \bar{T}_{O,j} = \sum_{j=1}^{h-1} (\kappa + 1)^j ((3 + (1 + \gamma)(2m_j - 1)) \xi T_G + (1 + \gamma) m_j T_{\text{pos},P}) . \quad (2.14)$$

Encore d'après le lemme 1, la complexité moyenne en temps sur l'ensemble des feuilles s'écrit :

$$\sum_{j=2}^{h-1} \bar{T}_{O,j} + 2\bar{T}_{O,m} = \sum_{j=2}^{h-1} (\kappa + 1)^{j-1} (3\xi T_G + \bar{T}_P) + 2(\kappa + 1)^{m-1} (3\xi T_G + \bar{T}_P) . \quad (2.15)$$

Le nombre d'objets primitifs constituant le sous-objet associé au nœud interne situé au niveau j est le nombre de feuilles du sous-arbre associé à ce nœud, on a donc :

$$m_j = m - (j - 1) .$$

En injectant l'expression de m_j dans 2.14 puis en séparant les différents termes on est amené à considérer les trois sommes suivantes :

$$\left\{ \begin{array}{l} \sum_{j=1}^{h-1} (\kappa + 1)^j = \frac{\kappa_0^h - \kappa_0}{\kappa_0 - 1} \\ \sum_{j=1}^{h-1} (\kappa + 1)^j j = \frac{\kappa_0^h (h \kappa_0 - \kappa_0 - h) + \kappa_0}{(\kappa_0 - 1)^2} \\ \sum_{j=2}^{h-1} (\kappa + 1)^{j-1} = \frac{\kappa_0^{h-1} - \kappa_0}{\kappa_0 - 1} \end{array} \right. .$$

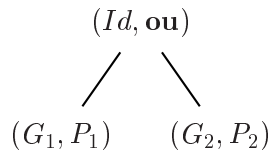
Il ne reste plus qu'à regrouper les différents termes obtenus à partir de 2.14 et 2.15 et de les injecter dans 2.13, puis de remarquer que la hauteur d'un arbre binaire dégénéré s'écrit $h = m$. On obtient alors l'expression proposée pour \bar{T}_O :

$$\begin{aligned} \bar{T}_O &= \frac{1}{(\kappa_0 - 1)^2} ((12 + 3\gamma) \kappa_0^{m+1} - (13 + \gamma) \kappa_0^m + 3 \kappa_0^{m-1} \\ &\quad - (6 + (2m - 1)(1 + \gamma)) \kappa_0^2 + (6 + (2m + 1)(1 + \gamma)) \kappa_0) \xi T_G \\ &\quad + (1 + \gamma) \frac{2 \kappa_0^{m+1} - \kappa_0^m - (m + 1) \kappa_0^2 + m \kappa_0}{(\kappa_0 - 1)^2} T_{\text{pos},P} + \frac{2 \kappa_0^m - \kappa_0^{m-1} - \kappa_0}{\kappa_0 - 1} \bar{T}_P . \end{aligned}$$

■

Exemple 4

Reprenons l'objet considéré dans l'exemple 3. Soit un objet O construit par union d'une sphère P_1 et d'un cylindre P_2 . Le nombre d'objets primitifs est donc $m = 2$. Un opérateur de translation G_1 est appliqué à la sphère tandis qu'un opérateur de rotation G_2 est appliqué au cylindre. Aucun opérateur géométrique n'est appliqué sur l'objet résultant de l'union.



Il y a 2 opérateurs géométriques dans l'arbre binaire associé à O , leur proportion est donc $\xi = (0 + 1 + 1)/3 = 2/3$. Supposons que l'ensemble \mathcal{R} est tel que le nombre moyen de surfaces virtuelles rencontrées par un rayon est $\kappa = 1$ et que la proportion moyenne de rayons qui ont une intersection est $\gamma = 1/2$. La valeur de la complexité moyenne en temps de la recherche de la première intersection entre un rayon issu de \mathcal{R} et l'objet O est donnée par :

$$\bar{T}_O = 18 T_G + 6 T_{\text{pos},P} + 4 \bar{T}_P .$$

Cette complexité est bien évidemment identique pour les deux types d'arbre considérés, puisque l'arbre représentant O est à la fois équilibré et dégénéré.

□

Pour comparer les complexités moyennes en temps dans les cas d'arbres binaires équilibrés et dégénérés, commençons par observer le tableau suivant.

	arbre équilibré	arbre dégénéré
$f_G(m, \kappa, \gamma)$	$O(3m(\kappa + 1)^{m-1})$	$O((12 + 3\gamma)(\kappa + 1)^{m-1})$
$f_{\text{pos},P}(m, \kappa, \gamma)$	$O(m(1 + \gamma)(\kappa + 1)^{\log_2(m)})$	$O(2(1 + \gamma)(\kappa + 1)^{m-1})$
$f_P(m, \kappa, \gamma)$	$O(m(\kappa + 1)^{m-1})$	$O(2(\kappa + 1)^{m-1})$

Ce tableau donne un ordre de grandeur des complexités moyennes en temps pour un nombre d'objets primitifs très grand. Cependant, il est à considérer avec la plus grande prudence. En effet, en pratique, le nombre d'objets primitifs considérés est loin d'être très grand, bien au contraire, les objets sont plutôt construits avec un petit nombre d'objets primitifs. Les courbes des fonctions intervenant dans les complexités considérées (figure 2.18) confirment cette remarque.

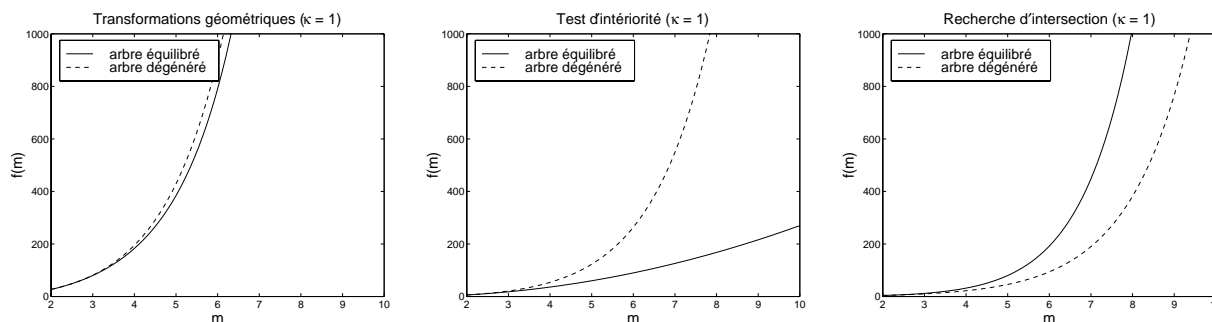


Fig. 2.18 – Courbes des fonctions intervenant dans la complexité moyenne en temps du calcul des transformations géométriques (f_G), du test d'intériorité ($f_{\text{pos},P}$) et de la recherche d'intersection (f_P).

Pour mieux comparer les deux types d'arbres, traçons les courbes des complexités moyennes en temps pour différentes valeurs de κ (figure 2.19). Pour ce faire il est nécessaire de se donner les valeurs ξ et γ et les valeurs des complexités en temps T_G , $T_{\text{pos},P}$ et \bar{T}_P . Le choix de ces valeurs n'est pas facile car il dépend de l'objet considéré, de la manière dont sont programmées les différentes méthodes considérées, du compilateur considéré, des paramètres de compilation utilisés et du matériel sur lequel sont exécutés ces programmes. On prendra donc des ordres de grandeur pour tracer ces courbes :

$$\left\{ \begin{array}{l} \xi = 1 \\ \gamma = 0.5 \\ T_G = 0.1 \\ T_{\text{pos},P} = 1 \\ \bar{T}_P = 10 \end{array} \right. .$$

Ces courbes montrent qu'il est préférable de construire des objets sous la forme d'un arbre dégénéré plutôt qu'un arbre équilibré. La complexité en temps ainsi obtenue est meilleure.

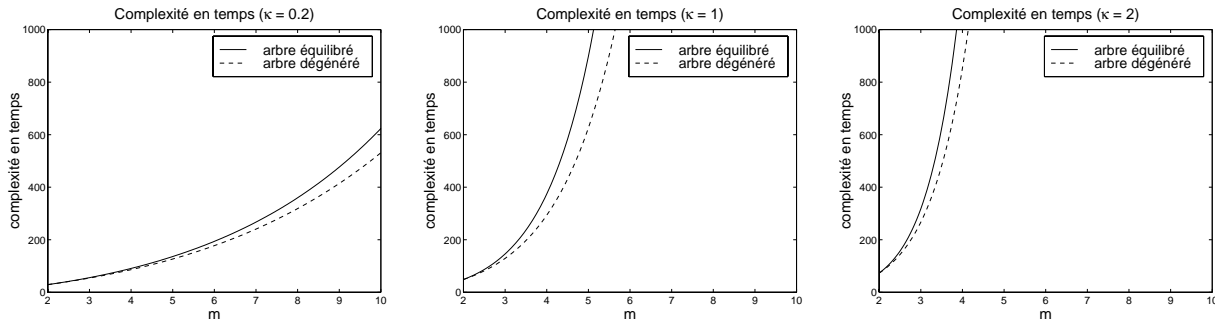


Fig. 2.19 – Complexité moyenne en temps pour différentes valeurs de κ (\overline{T}_O).

Méthode de Roth

La proposition suivante donne une expression récursive de la complexité en temps pour la méthode de Roth (algorithme 4).

Proposition 8

Soit $R_{x,w}$ un rayon. Notons $T_P(x, w)$ la complexité en temps de recherche de la première intersection entre le rayon et l'objet primitif P . Notons T_G la complexité en temps relative au calcul de la transformée d'un point ou d'un vecteur par un opérateur géométrique. Notons $\lambda_i(x, w)$ le nombre de points d'intersection entre le rayon considéré et un objet O_i . Notons $T_E(\lambda_1(x, w), \dots, \lambda_n(x, w))$ la complexité en temps pour évaluer la liste d'intervalles d'intersection résultante de l'application de l'opérateur E à n listes de tailles respectives $\lambda_1(x, w), \dots, \lambda_n(x, w)$. La complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et un objet O s'exprime de manière récursive :

$$T_O(x, w) = \begin{cases} T_P(x, w) & \text{si } O = P \\ (2 + \lambda_1(x', w')) T_G + T_{O_1}(x', w') & \text{si } O = (G, O_1) \\ T_E(\lambda_1(x, w), \dots, \lambda_n(x, w)) + \sum_{i=1}^n T_{O_i}(x, w) & \text{si } O = (E, O_1, \dots, O_n) \end{cases},$$

où x' et w' sont les paramètres du rayon $G^{-1}(R_{x,w})$.

□

Preuve Soit $R_{x,w}$ un rayon.

Si O est un objet primitif P (y compris les objets construits à partir d'opérateurs géométriques) alors la complexité en temps est propre à l'algorithme de calcul d'intersection du rayon avec P et on a :

$$T_O(x, w) = T_P(x, w) .$$

Si $O = (G, O_1)$ où G est un opérateur géométrique, alors il faut décomposer la méthode. Dans un premier temps, le rayon $R_{x,w}$ est transformé à l'aide de l'opérateur G^{-1} :

$$R_{x',w'} = G^{-1}(R_{x,w}) .$$

La complexité en temps associée est $2T_G$ (il faut transformer les deux paramètres du rayon). Ensuite, la recherche de liste d'intervalles d'intersection entre le rayon $R_{x',w'}$ et l'objet O_1 est effectuée. La complexité en temps associée est $T_{O_1}(x', w')$. Enfin, la liste d'intervalles d'intersection recherchée, si elle existe, doit être replacée dans le repère de O , ce qui s'obtient par application de l'opérateur géométrique G . La complexité en temps associée est $\lambda_1(x', w') T_G$, où $\lambda_1(x', w')$ est le nombre d'éléments

de la liste considérée, c'est-à-dire le nombre de points d'intersection entre $R_{x',w'}$ et O_1 . La complexité en temps totale pour la recherche de la première intersection entre le rayon $R_{x,w}$ et l'objet O s'écrit donc de manière récursive :

$$T_O(x, w) = (2 + \lambda_1(x', w')) T_G + T_{O_1}(x', w') .$$

Si $O = (E, O_1, \dots, O_n)$ où E est un opérateur ensembliste, alors la méthode consiste à rechercher les listes d'intervalles d'intersection avec tous les objets O_i , puis à évaluer la liste résultante en fonction de l'opérateur E . La complexité en temps associée s'écrit donc :

$$T_O(x, w) = T_E(\lambda_1(x, w), \dots, \lambda_n(x, w)) + \sum_{i=1}^n T_{O_i}(x, w) .$$

D'où l'expression de la complexité en temps proposée. ■

Remarque

La complexité $T_E(\lambda_1(x, w), \dots, \lambda_n(x, w))$ n'est pas très explicite de manière générale. Cependant, dans le cas où $n = 2$ son expression est la suivante :

$$T_E(\lambda_1(x, w), \lambda_2(x, w)) = \frac{\lambda_1(x, w) + \lambda_2(x, w)}{2} T_{\text{liste}} ,$$

où T_{liste} est la complexité en temps du traitement des opérations de comparaison et de construction de la liste résultante. Pour s'en convaincre il suffit de remarquer que le tri de deux listes d'intervalles d'intersection, quel que soit l'opérateur ensembliste considéré, nécessite le parcours des intervalles de ces deux listes, donc de $\lambda_i(x, w)/2$ nombres pour chaque liste. □

Corollaire 2

La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et un objet O s'écrit de la manière suivante :

$$\bar{T}_O = \begin{cases} \bar{T}_P & \text{si } O = P \\ (2 + \bar{\lambda}_1) T_G + \bar{T}_{O_1} & \text{si } O = (G, O_1) \\ T_E(\bar{\lambda}_1, \dots, \bar{\lambda}_n) + \sum_{i=1}^n \bar{T}_{O_i} & \text{si } O = (E, O_1, \dots, O_n) \end{cases} .$$

□

Preuve Il suffit de partir de la définition 1 de la complexité moyenne en temps par rapport à l'ensemble de rayons issus de \mathcal{R} à l'expression de la complexité de la proposition 8. On remarquera que l'on peut bien écrire $T_E(\bar{\lambda}_1, \dots, \bar{\lambda}_n)$. En effet, d'après la remarque précédente, l'expression $T_E(\lambda_1(x, w), \dots, \lambda_n(x, w))$ est linéaire en fonction des $\lambda_i(x, w)$, ce qui autorise la décomposition de l'intégrale au numérateur de l'expression de la complexité moyenne en temps. ■

L'expression récursive du corollaire 2 n'est pas très explicite, c'est pourquoi on propose de donner une expression de la complexité moyenne en temps de la méthode de Roth qui soit fonction des objets primitifs considérés, et des opérateurs rencontrés. Pour ce faire, on utilise la proposition 1 qui permet de travailler avec un arbre binaire. De manière analogue à la méthode des epsilons, on commence par donner un lemme qui utilisé dans la démonstration de la proposition 9.

Lemme 2

Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m . Notons \overline{T}_P la complexité moyenne en temps de recherche de la première intersection moyennée sur l'ensemble des objets primitifs :

$$\overline{T}_P = \frac{1}{m} \sum_{i=1}^m \overline{T}_{P_i} .$$

Notons λ le nombre moyen de points d'intersection entre un rayon issu de \mathcal{R} et un sous-objet de O . Notons T_G la complexité en temps relative au calcul de la transformée d'un point ou d'un vecteur par un opérateur géométrique. Notons ξ la proportion d'éléments de l'arbre qui sont associés à un opérateur géométrique qui n'est pas l'identité. Notons T_{liste} la complexité en temps du traitement des opérations de comparaison et de construction lors de l'évaluation d'une liste d'intervalles d'intersection. La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et le sous-objet associé à un élément situé au niveau j de l'arbre de O s'écrit :

$$\overline{T}_{O,j} = \begin{cases} (2 + \lambda) \xi T_G + \lambda T_{\text{liste}} & \text{si nœud interne} \\ (2 + \lambda) \xi T_G + \overline{T}_P & \text{si feuille} \end{cases} .$$

□

Preuve Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m (y compris les objets construits à partir d'opérateurs constructifs). L'arbre CSG associé à O est par hypothèse un arbre binaire.

Dire que ξ est la proportion d'éléments de l'arbre qui sont associés à un opérateur géométrique qui n'est pas l'identité, c'est dire que la complexité en temps associée à chaque élément de l'arbre pour les opérateurs géométriques s'écrit $(2 + \lambda) \xi T_G$. Le facteur $2 + \lambda$ provient du nombre de transformations géométriques à effectuer, il y en a deux pour transformer un rayon et λ pour transformer une liste d'intervalles d'intersection contenant λ éléments (une justification précise est donnée dans la démonstration de la proposition 8). Par définition de ξ , ξT_G est la complexité en temps moyennée sur les éléments de l'arbre (une justification précise est donnée dans la démonstration de la proposition 5). Un nœud interne situé au niveau j dans l'arbre contient un opérateur géométrique et un opérateur ensembliste. Il est nécessaire d'évaluer la liste d'intervalles d'intersection résultante. Par hypothèse, l'arbre est binaire. De plus, le nombre moyen de points d'intersection λ est le même pour tous les nœuds. La complexité en temps de l'évaluation de la liste d'intervalles d'intersection résultante s'écrit donc :

$$T_E(\lambda, \lambda) = \frac{\lambda + \lambda}{2} T_{\text{liste}} = \lambda T_{\text{liste}} .$$

La complexité moyenne en temps associée à un tel nœud est donc :

$$\overline{T}_{O,j} = (2 + \lambda) \xi T_G + \lambda T_{\text{liste}} .$$

Une feuille située au niveau j de l'arbre CSG contient un opérateur géométrique et un objet primitif. Il est nécessaire d'effectuer le calcul d'intersection entre un rayon et l'objet primitif considéré. La complexité moyenne en temps associée à un telle feuille est donc :

$$\overline{T}_{O,j} = (2 + \lambda) \xi T_G + \overline{T}_P .$$

■

Remarque

Le nombre moyen de point d'intersection λ est moyenné sur l'ensemble \mathcal{R} , et devrait en toute rigueur être noté $\overline{\lambda}$. C'est par souci de clarté dans les écritures que l'on a adopté cette notation.

□

Proposition 9

Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m . Considérons les mêmes notations qu'au lemme 2. La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et l'objet O s'écrit :

$$\overline{T}_O = f_G(m, \lambda) \xi T_G + f_{\text{liste}}(m, \lambda) T_{\text{liste}} + f_P(m, \lambda) \overline{T}_P, \quad (2.16)$$

où les fonctions f_G , f_{liste} et f_P sont définies par :

$$\begin{cases} f_G(m, \lambda) & = (2m - 1)(2 + \lambda) \\ f_{\text{liste}}(m, \lambda) & = (m - 1)\lambda \\ f_P(m, \lambda) & = m \end{cases} .$$

□

Preuve Soit O un objet CSG construit à partir de m objets primitifs P_1, \dots, P_m (y compris les objets construits à partir d'opérateurs constructifs). La proposition 1 précise que l'objet CSG O peut être représenté par un arbre binaire.

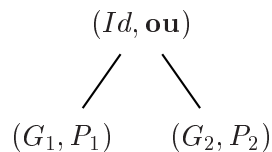
Pour évaluer la complexité moyenne en temps associée à tous les éléments de l'arbre, ajoutons les complexités moyennes aux nœuds de l'arbre. D'après le lemme 2, les complexités relatives aux nœuds situés au niveau j sont indépendantes de j . L'arbre binaire associé à O contient $m - 1$ nœuds internes et m feuilles. Par conséquent, la complexité moyenne en temps recherchée s'écrit :

$$\begin{aligned} \overline{T}_O &= (m - 1) ((2 + \lambda) \xi T_G + \lambda T_{\text{liste}}) + m ((2 + \lambda) \xi T_G + \overline{T}_P) \\ &= (2m - 1)(2 + \lambda) T_G + (m - 1) \lambda T_{\text{liste}} + m \overline{T}_P . \end{aligned}$$

■

Exemple 5

Reprenons l'objet considéré dans les exemples 3 et 4. Soit un objet O construit par union d'une sphère P_1 et d'un cylindre P_2 . Le nombre d'objets primitifs est donc $m = 2$. Un opérateur de translation G_1 est appliqué à la sphère tandis qu'un opérateur de rotation G_2 est appliqué au cylindre. Aucun opérateur géométrique n'est appliqué sur l'objet résultant de l'union.



Il y a 2 opérateurs géométriques dans l'arbre binaire associé à O , leur proportion est donc $\xi = (0 + 1 + 1)/3 = 2/3$. Supposons que l'ensemble \mathcal{R} est tel que le nombre moyen d'intersections rencontrées par un rayon est $\lambda = 2$. La valeur de la complexité moyenne en temps de la recherche de la première intersection entre un rayon issu de \mathcal{R} et l'objet O est donnée par :

$$\overline{T}_O = 12 T_G + 2 T_{\text{liste}} + 2 \overline{T}_P .$$

□

Comparaison des deux méthodes

Pour comparer les méthodes des epsilons et de Roth, traçons les courbes des complexités moyennes en temps de chaque méthode en faisant varier κ et λ (figures 2.20 et 2.21). Pour ce faire, il est nécessaire de se donner les valeurs de ξ et γ et les valeurs des complexités en temps T_G , $T_{\text{pos},P}$, T_{liste} et T_P . Le choix de ces valeurs n'est pas facile car il dépend de beaucoup de paramètres. Pour donner une idée dans un cadre général considérerons des ordres de grandeurs :

$$\left\{ \begin{array}{l} \xi = 1 \\ \gamma = 0.5 \\ T_G = 0.1 \\ T_{\text{pos},P} = 1 \\ T_{\text{liste}} = 1 \\ \overline{T}_P = 10 \end{array} \right. .$$

Il est aussi nécessaire de discuter du choix des paramètres κ et λ . Soit un rayon issu de \mathcal{R} et un sous-objet de l'objet considéré. κ est le nombre de surfaces virtuelles rencontrées lors de la recherche de la première intersection entre le rayon et le sous-objet considérés dans la méthode des epsilons. λ est le nombre de points d'intersection entre le rayon et le sous-objet considéré dans la méthode de Roth. La seule relation particulière (relation 2.7) est que κ est majoré par la somme des nombres de points d'intersection des sous-objets. Puisqu'on est dans le cas d'un arbre binaire, on a la relation suivante :

$$\kappa \leq 2\lambda .$$

Pour les courbes, on prendra donc deux cas de figure :

- κ constant ($\kappa = 0$), λ varie,
- κ et λ varient.

De plus, pour la méthode des epsilons, la complexité moyenne en temps choisie est la plus favorable, c'est-à-dire le cas d'un arbre binaire dégénéré. Pour le cas $\kappa = 0$, l'équation 2.9 est toujours valable, mais les expressions des fonctions f_G , $f_{\text{pos},P}$ et f_P données dans la proposition 7 ne le sont plus. Dans ce cas particulier, la complexité moyenne en temps pour un arbre binaire dégénéré est donnée par un calcul identique à celui de la démonstration de la proposition 7 :

$$\left\{ \begin{array}{l} f_G(m, \kappa, \gamma) = (1 + \gamma) m^2 + 6m - (4 + \gamma) \\ f_{\text{pos},P}(m, \kappa, \gamma) = \frac{1}{2} (1 + \gamma) m^2 + \frac{1}{2} (1 + \gamma) m - (1 + \gamma) \\ f_P(m, \kappa, \gamma) = m \end{array} \right. .$$

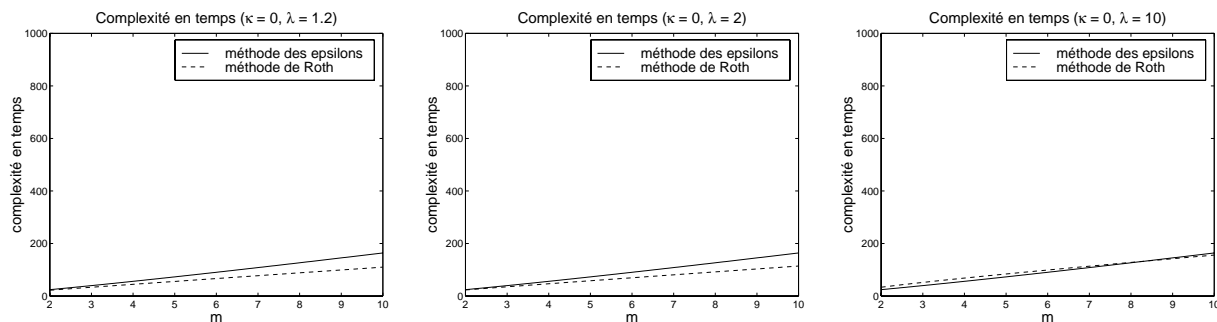


Fig. 2.20 – Comparaison des complexités en temps pour les méthodes des epsilons et de Roth (κ constant, λ varie).

Le premier cas (κ constant, λ varie) est le cas idéal pour la méthode des epsilons. En effet, $\kappa = 0$ signifie qu'il n'y a pas de répétitions dans les calculs d'intersection, chaque objet primitif n'est rencontré qu'une seule fois, ce qui est confirmé par $f_P(m, \kappa) = m$. Les courbes 2.20 montrent qu'il faut prendre des valeurs de λ grandes par rapport à κ pour que la complexité moyenne en temps de méthode des epsilons soit équivalente à celle de la méthode de Roth.

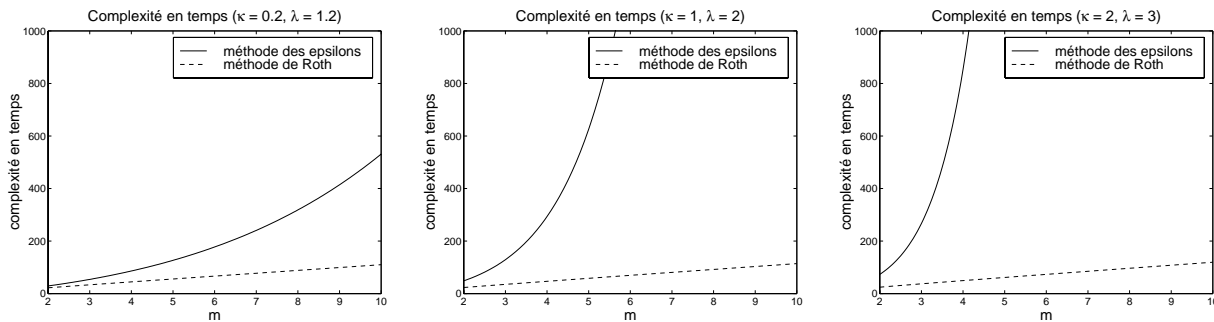


Fig. 2.21 – *Comparaison des complexités en temps pour les méthodes des epsilons et de Roth (κ et λ varient).*

Le second cas montre la supériorité de la méthode de Roth lorsque les paramètres κ et λ ont la même valeur. Les valeurs des complexités moyennes en temps données dans les exemples 4 et 5 se retrouvent au niveau de l'axe des ordonnées ($m = 2$) sur le graphe du milieu ($\kappa = 1, \lambda = 2$).

2.3 Modèle géométrique BRep

Le modèle géométrique de type BRep (*Boundary Representation*) définit une représentation par frontières des objets. Un objet est décrit par une frontière (modélisation surfacique) et par des relations visant à garantir une certaine cohérence topologique. Les différents modèles géométriques de type BRep se différencient par le type de surfaces utilisées et par l'étendue de la topologie considérée.

La description topologique d'un objet consiste donc à définir des volumes, des faces, des arêtes, des sommets et des liens entre ces éléments de la frontière (incidence, contiguïté). Les modèles les plus simples se contentent d'une topologie eulérienne (le voisinage de tout point de la frontière est homéomorphe à un disque), les plus complexes admettent des topologies non eulériennes voire hétérotopes (représentation pertinente des objets ponctuels, linéiques, surfaciques et volumiques avec données et procédures spécifiques et mieux adaptées).

Du point de vue géométrique, les modèles les plus simples approchent les surfaces par des faces planes, les modèles les plus complexes admettent des surfaces paramétrées de type Bézier, spline ou encore NURBS, éventuellement restreintes ou des surfaces implicites.

L'importance de la modélisation géométrique dans le monde industriel est fortement croissante. Les éditeurs de logiciels de CAO sont nombreux et, de ce fait, il existe de nombreux formats de données pour représenter les différents modèles géométriques existants. Les normes relatives aux fichiers de données CAO qui se mettent en place actuellement favorisent la représentation par frontière. D'autre part, la connaissance explicite des frontières permet de visualiser les objets en temps réel, fournissant ainsi aux utilisateurs une interactivité très bénéfique.

Cependant, la représentation explicite des surfaces est très gourmande en espace mémoire, de plus toute l'information représentée n'est pas toujours indispensable. De plus, il y a des problèmes de cohérence à cause de l'imprécision numérique. Ce problème de robustesse est un sujet de recherches activement étudié par la communauté CAO. Enfin, l'usage des opérations booléennes n'est pas sans difficulté. La complexité des structures de données, l'imprécision numérique et les nombreux cas particuliers rendent la mise en place de ces opérations ensemblistes très délicate.

Dans ce paragraphe, on ne s'intéressera pas aux algorithmes constructifs de la représentation BRep.

2.3.1 Description de la topologie

D'un point de vue topologique, un objet O de type BRep est décrit par :

- un ensemble de volumes $\{V_i\}_i$,
- un ensemble de faces $\{F_i\}_i$,
- un ensemble d'arêtes $\{A_i\}_i$,
- un ensemble de sommets $\{S_i\}_i$,
- un ensemble de liens entre les volumes, faces, arêtes et sommets.

Parmi les différentes représentations topologiques, on distingue deux modèles :

- le modèle arborescent,
- le modèle combinatoire.

Modèle arborescent

Les modèles hiérarchiques sont les plus répandus dans le domaine de la CAO. La représentation d'un objet de type BRep consiste en la description des sous-objets le composant, les relations topologiques étant exprimées de manière descendante par rapport à la dimension des sous-objets (figure 2.22).

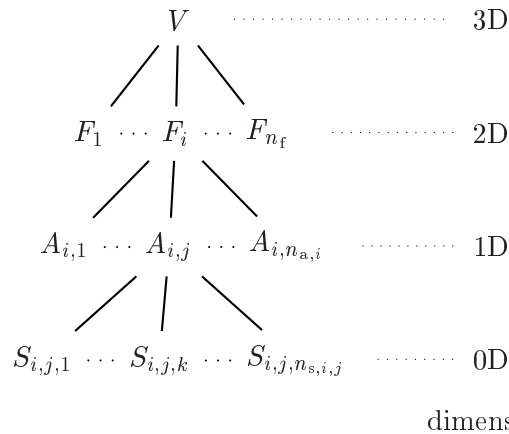


Fig. 2.22 – Représentation arborescente relativement à la dimension.

Cette représentation topologique est assez générale, et peut prendre en compte des topologies non eulériennes. Cependant, un des grands handicaps est la recherche des éléments de dimension supérieure contenant une entité géométrique donnée. Par exemple, étant donnée une arête, pour retrouver les faces qui la contiennent, il faut parcourir l'arbre considéré de haut en bas. Un autre inconvénient de taille est la duplication de l'information au sein de la structure de données, qui pose des problèmes de mise à jour et d'encombrement mémoire.

Modèle combinatoire

Les modèles combinatoires ont pour but de décrire des liens entre les différentes entités géométriques voisines, non pas au sens de la dimension, mais au sens des relations d'incidence (relation entre deux entités de dimensions différentes) et d'adjacence (relation entre deux entités de même dimension). Parmi ces modèles on peut citer le modèle DCEL (*Direct Connected Edge List*) (Carlier et Szafran [14] et Degott [28]) dont la définition des liens est basée sur les arêtes (figure 2.23). Cependant, ce modèle

ne prend en compte que des topologies eulériennes, c'est-à-dire que les nombres de faces, d'arêtes et de sommets sont liés par la relation d'Euler. De plus, c'est un modèle qui ne peut pas prendre en compte des entités géométriques 3D (au sens volumique).

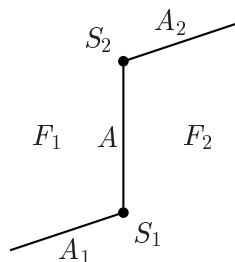


Fig. 2.23 – Structure DCEL.

Pour pallier cet inconvénient, un modèle plus général, appelé DCEL généralisé, a été développé (Degott, Chenin et Dessarce [29]). Il peut prendre en compte des topologies non eulériennes et des entités géométriques 3D. Cependant, la nature de ce modèle pose des problèmes de gestion et de mise à jour des structures de données associées.

Le modèle des cartes combinatoires (Lienhardt [60] [61]) est très attrayant par son formalisme mathématique fondé sur la théorie des groupes de permutation. Il est défini à partir d'un type unique d'élément, appelé brin, sur lequel sont définis des opérateurs de permutation. Un brin correspond à une arête orientée. Les notions de cellules, de composantes connexes et de bords sont définis par l'intermédiaire de la notion d'orbite. Ce modèle utilise des opérateurs algébriques pour définir les relations de voisinages et de cohérence topologique. Il peut prendre en compte des topologies non eulériennes et permet, entre autres, de modéliser des variétés cellulaires⁷ orientables de dimension quelconque. Le modèle des cartes combinatoires généralisées permet en plus de considérer des variétés cellulaires non orientables.

Les modèles combinatoires ont pour principal avantage de permettre un accès direct aux différentes entités géométriques, et donc d'une manière beaucoup plus rapide que dans le modèle hiérarchique, en contrepartie d'une complexité en mémoire importante.

2.3.2 Description de la géométrie

D'un point de vue géométrique, un objet BRep est défini en associant à chaque face une surface (supposée continue, éventuellement plus, sans auto-intersection), décrite de manière paramétrique ou implicite. De manière similaire, à chaque arête est associée une courbe (supposée continue, éventuellement plus, sans auto-intersection), décrite de manière paramétrique ou implicite.

Contrairement au modèle CSG, où la nature des objets primitifs dépend du contexte, ici la nature des surfaces et des courbes utilisées est plutôt liée à une normalisation des échanges de données issues de la CAO (IGES, STEP). Dans le contexte de l'optique, il faut remarquer que la construction d'un système optique s'effectue en deux phases. La première phase a pour but de définir la géométrie du système optique, tandis que la seconde consiste à associer les propriétés optiques aux différentes entités de cette géométrie. Les entités géométriques considérées sont donc celles que l'on retrouve classiquement dans le domaine de la mécanique. Dans le contexte qui nous intéresse, on classe ces entités géométriques en trois catégories :

- les objets de base définis de manière paramétrique ou implicite (figure 2.24),
- les objets paramétriques polynômiaux (figure 2.25),
- les objets définis à partir d'un opérateur constructif (figure 2.25).

⁷La notion de variété cellulaire est une notion générale qui englobe la notion de surface dans \mathbb{R}^3 . Toute surface dans \mathbb{R}^3 est triangularisable et peut donc être représentée par une variété cellulaire. La notion de variété permet d'introduire des structures combinatoires.

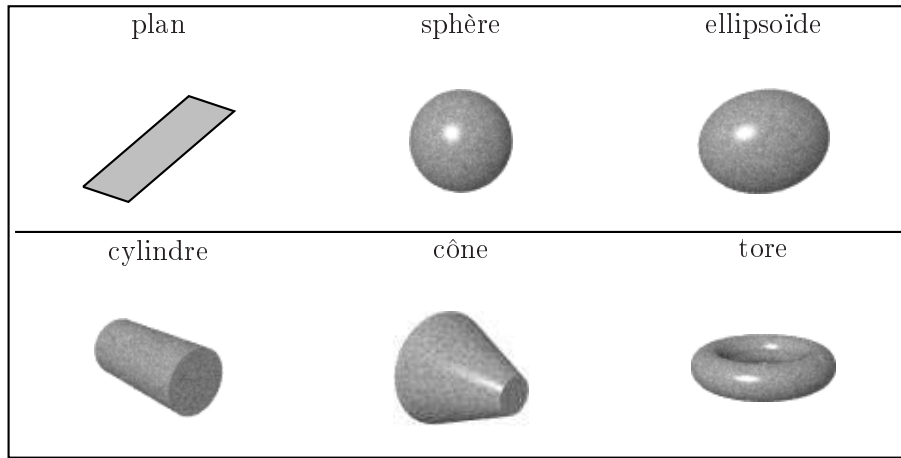


Fig. 2.24 – Objets géométriques de base.

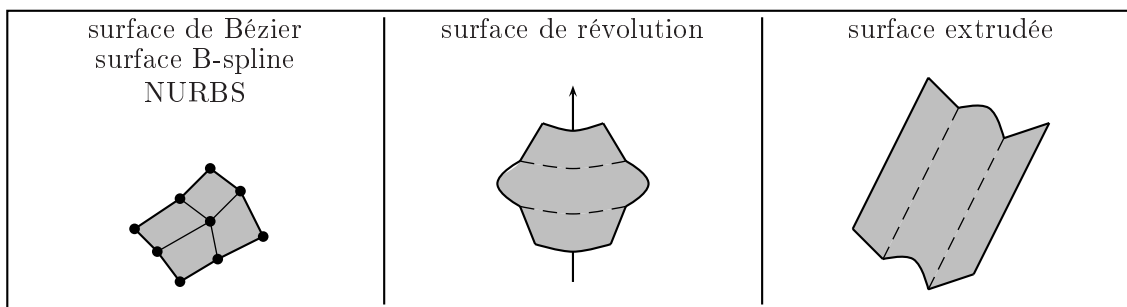


Fig. 2.25 – Objets paramétriques polynômiaux et objets définis à partir d'un opérateur constructif (surface de révolution et surface extrudée).

2.3.3 Utilisation du modèle pour l'optique

En optique la mise en place d'une simulation nécessite la représentation de la fonction indice de réfraction optique⁸ $\eta : X \subset \mathbb{R}^3 \rightarrow \mathbb{R}$. De façon analogue au cas CSG, les objets peuvent être classés en deux catégories par rapport aux propriétés optiques :

- les objets susceptibles d'émettre des photons,
- les objets qui définissent une discontinuité de la fonction η .

Un objet appartient à l'une, l'autre ou les deux catégories. Toutes les entités géométriques d'un modèle BRep peuvent être susceptibles d'émettre des photons. Concernant la deuxième catégorie d'objets, il faut être plus précis par rapport au contexte. Le modèle topologique considéré est de type hiérarchique et ne prend pas en compte les entités géométriques volumiques. Un objet BRep est par définition représenté par son contour, c'est-à-dire par un ensemble de faces. La notion de volume ne peut pas être considérée à cause, d'une part, des problèmes d'imprécision numérique qui conduisent à des trous entre les faces ou à des faces ou des arêtes pendantes, et, d'autre part, de l'instabilité des opérateurs ensemblistes. C'est pourquoi les notions d'intérieur et d'extérieur à un objet n'ont pas de sens robuste dans ce modèle. Il n'est donc pas envisageable de représenter la fonction η constante par morceaux, autrement dit de considérer des changements de milieu. L'information doit donc être portée par les faces, qui représentent une discontinuité de la fonction η . Si l'on veut définir un changement de milieu, il faut alors orienter les faces et considérer une valeur pour η de chaque côté de la face considérée. Il est nécessaire d'assurer en même temps une cohérence topologique (orientation des faces cohérente

⁸Voir la définition donnée au chapitre 1 page 15.

pour un même objet) et une cohérence géométrique (valeurs de η cohérentes pour des faces contiguës à un même milieu). Les arêtes et les sommets peuvent aussi jouer un rôle en optique. Bien que n'étant pas pris en compte dans les équations du premier chapitre, les phénomènes de diffraction peuvent être envisagés.

Dans la suite, on suppose disposer d'une structure de données cohérente. La fonction essentielle qui nous intéresse est le calcul d'intersection entre un rayon et un objet.

2.3.4 Intersection entre un rayon et un objet

Soit O un objet de type BRep. Sans aucune hypothèse sur l'objet, et si l'on ne considère que les surfaces optiques (on ne tient pas compte des arêtes et des sommets), alors la recherche de la première intersection entre un rayon $R_{x,w}$ et la surface décrite par l'objet O consiste à rechercher la première intersection avec l'ensemble des faces $\{F_i\}_{i=1}^n$. L'algorithme correspondant est trivial (algorithme 5).

Algorithme 5: *Intersection entre un rayon et un objet BRep.*

```

INTERRAYONOBJETBREP( $x, w, O$ )
(1)  ( $t_{\min}, I_{\min}$ )  $\leftarrow$  ( $+\infty, \{\}$ )
(2)  for  $i = 1$  to  $n$ 
(3)    ( $t, I$ )  $\leftarrow$  INTERRAYONFACE( $x, w, F_i$ )
(4)    if  $t < t_{\min}$ 
(5)      ( $t_{\min}, I_{\min}$ )  $\leftarrow$  ( $t, I$ )
(6)  return ( $t_{\min}, I_{\min}$ )

```

L'algorithme INTERRAYONFACE calcule l'intersection entre un rayon et la surface associée à la face considérée. Si une intersection est trouvée, alors il renvoie le paramètre correspondant sur le rayon et, éventuellement, diverses informations relatives à l'objet et au point d'intersection trouvé (indice de la face, point dans l'espace des paramètres pour un objet paramétrique, etc). Sinon, par convention, il renvoie ($+\infty, \{\}$).

L'algorithme 5 est coûteux en ce sens qu'il ne prend en compte aucune propriété de l'objet considéré. Ainsi, pour certains types d'objets, il peut être très avantageux de prendre en compte la structure de données pour le parcours des faces. Par exemple, on peut ne pas tester toutes les faces à l'aide de critères de localisation géométrique (souvent ce sont des boîtes englobantes de faces).

Si l'on souhaite tenir compte des arêtes et/ou des sommets, alors il faut adapter l'algorithme 5, et considérer des intersections entre des photons et des arêtes et/ou sommets. Numériquement, l'intersection entre un rayon et une arête ou un sommet ne peut être envisagée, c'est pourquoi on se donne une tolérance qui consiste à prendre en compte un voisinage de l'arête ou du sommet. Généralement cela revient à considérer un cylindre ou un parallélépipède autour de l'arête considérée et une sphère ou un cube autour du sommet considéré.

2.3.5 Complexité en mémoire

Dans ce paragraphe, on cherche à évaluer la complexité en mémoire d'une scène géométrique de type BRep, c'est-à-dire d'une liste d'objets BRep O^1, \dots, O^N . Cette complexité représente la place pour stocker toutes les données définissant cette liste d'objets, on la note M_{O^1, \dots, O^N} , et on a la relation suivante :

$$M_{O^1, \dots, O^N} = \sum_{k=1}^N M_{O^k}, \quad (2.17)$$

où M_{O^k} est la complexité en mémoire pour l'objet O^k . Pour calculer cette complexité il est donc nécessaire d'évaluer la complexité en mémoire d'un objet BRep. Elle dépend du modèle topologique

et des types d'entités géométriques considérés. De manière générale, pour un objet BRep O cette complexité en mémoire s'écrit :

$$M_O = M_{\text{topo},O} + M_{\text{geo},O} ,$$

où $M_{\text{topo},O}$ et $M_{\text{geo},O}$ sont les complexités en mémoire respectivement pour la topologie et pour la géométrie. Dans la suite, pour être conforme au contexte qui nous concerne, on considère un objet BRep représentant une surface, c'est-à-dire décrit par :

- un ensemble de faces $\{F_i\}_{i=1}^n$,
- un ensemble d'arêtes $\{A_{i,j}\}_{i,j}$,
- un ensemble de sommets $\{S_{i,j,k}\}_{i,j,k}$,

et dont le modèle de topologie est de type hiérarchique. Les entités géométriques sont usuellement décrites par des enregistrements et leurs relations topologiques par des références (des pointeurs ou des indices dans des tables) entre ces enregistrements. On supposera qu'une telle référence équivaut à une unité de complexité en mémoire. La proposition suivante donne une expression de la complexité en mémoire relative à la topologie d'un objet BRep.

Proposition 10

Soit O un objet BRep. La complexité en mémoire liée à la topologie de l'objet O s'écrit de la manière suivante :

$$M_{\text{topo},O} = \sum_{i=1}^n \left(n_{F_i} + \sum_{j=1}^{n_{F_i}} n_{A_{i,j}} \right) , \quad (2.18)$$

où n_{F_i} et $n_{A_{i,j}}$ sont les nombres respectivement d'arêtes liées à la face F_i et de sommets liés à l'arête $A_{i,j}$. □

Preuve Soit O un objet de type BRep. Dans le modèle hiérarchique, une face F_i est liée à n_{F_i} arêtes et une arête $A_{i,j}$ est liée à $n_{A_{i,j}}$ sommets. Un lien est représenté par une référence à l'entité géométrique considérée, par convention c'est une unité de complexité en mémoire. En comptabilisant les faces et les arêtes on a bien la relation proposée. ■

La proposition suivante donne une expression de la complexité en mémoire relative à la géométrie d'un objet BRep.

Proposition 11

Soit O un objet BRep. Notons M_{F_i} , $M_{A_{i,j}}$ et $M_{S_{i,j,k}}$ les complexités en mémoire relatives à la géométrie respectivement de la face F_i , de l'arête $A_{i,j}$ et du sommet $S_{i,j,k}$. La complexité en mémoire relative à la géométrie de l'objet O s'écrit de la manière suivante :

$$M_{\text{geo},O} = \sum_{i=1}^n \left(M_{F_i} + \sum_{j=1}^{n_{F_i}} \left(M_{A_{i,j}} + \sum_{k=1}^{n_{A_{i,j}}} M_{S_{i,j,k}} \right) \right) , \quad (2.19)$$

où n_{F_i} et $n_{A_{i,j}}$ sont les nombres respectivement d'arêtes liées à la face F_i et de sommets liés à l'arête $A_{i,j}$. □

Preuve Il suffit de comptabiliser les différentes entités géométriques considérées. ■

Exemple 6

Considérons un objet BRep O composé de n_{face} faces triangulaires. Une face est liée à trois arêtes et une arête à deux sommets. La complexité en mémoire liée à la topologie de l'objet O s'écrit donc de la manière suivante :

$$M_{\text{topo},O} = \sum_{i=1}^{n_{\text{face}}} \left(3 + \sum_{j=1}^3 2 \right) = 9 n_{\text{face}} .$$

Une face est représentée par trois sommets, une arête par deux sommets et un sommet par trois nombres réels. Considérons qu'un nombre réel nécessite une unité de complexité en mémoire. La complexité en mémoire liée à la géométrie de l'objet O s'écrit donc :

$$M_{\text{geo},O} = \sum_{i=1}^{n_{\text{face}}} \left((3 + 3 + 3) + \sum_{j=1}^3 \left((3 + 3) + \sum_{k=1}^2 3 \right) \right) = 45 n_{\text{face}} .$$

La complexité en mémoire de l'objet O s'écrit finalement :

$$M_O = 54 n_{\text{face}} .$$

□

Exemple 7

Considérons un objet BRep O composé de n_{face} faces de type carreau polynomial paramétrique bi-cubique. Une face est liée à quatre arêtes et une arête à deux sommets. La complexité en mémoire liée à la topologie de l'objet O s'écrit donc de la manière suivante :

$$M_{\text{topo},O} = \sum_{i=1}^{n_{\text{face}}} \left(4 + \sum_{j=1}^3 2 \right) = 10 n_{\text{face}} .$$

Un carreau polynomial paramétrique bi-cubique est représenté par seize points dans \mathbb{R}^3 , une arête est une courbe polynomiale de degré 3, elle est donc représentée par quatre coefficients réels et un sommet est représenté par trois nombres réels. Considérons qu'un nombre réel nécessite une unité de complexité en mémoire. La complexité en mémoire liée à la géométrie de l'objet O s'écrit donc :

$$M_{\text{geo},O} = \sum_{i=1}^{n_{\text{face}}} \left((16 * 3) + \sum_{j=1}^3 \left(4 + \sum_{k=1}^2 3 \right) \right) = 78 n_{\text{face}} .$$

La complexité en mémoire de l'objet O s'écrit finalement :

$$M_O = 88 n_{\text{face}} .$$

□

2.3.6 Complexité en temps

Dans ce paragraphe, on cherche à évaluer la complexité en temps de recherche de la première intersection entre un rayon $R_{x,w}$ et une scène géométrique de type BRep, c'est-à-dire une liste d'objets BRep O^1, \dots, O^N , pour une méthode naïve (algorithme 6 page 90). Cette complexité représente le coût de l'algorithme qui retourne, s'il existe, le point d'intersection recherché, on la note T_{O^1, \dots, O^N} , et on a la relation suivante (proposition 13 page 90) :

$$T_{O^1, \dots, O^N}(x, w) = \sum_{k=1}^N T_{O^k}(x, w) , \quad (2.20)$$

où $T_{O^k}(x, w)$ est la complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et l'objet O^k . Il faut donc évaluer la complexité en temps pour un objet BRep. De manière analogue au paragraphe précédent, on considère un objet BRep représentant une surface, c'est-à-dire décrit par :

- un ensemble de faces $\{F_i\}_{i=1}^n$,
- un ensemble d'arêtes $\{A_{i,j}\}_{i,j}$,
- un ensemble de sommets $\{S_{i,j,k}\}_{i,j,k}$,

et dont le modèle de topologie est de type hiérarchique. Dans le contexte qui nous intéresse (algorithme 5), la topologie n'est pas prise en compte et seules les faces sont considérées. La proposition suivante donne une expression de la complexité en temps de recherche de la première intersection entre un rayon et un objet BRep.

Proposition 12

Soit $R_{x,w}$ un rayon. La complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et un objet O s'écrit de la manière suivante :

$$T_O(x, w) = \sum_{i=1}^n T_{F_i}(x, w) , \quad (2.21)$$

où $T_{F_i}(x, w)$ est la complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et la face F_i . □

Preuve La lecture de l'algorithme 5 donne immédiatement l'expression proposée. ■

Corollaire 3

La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et un objet O s'écrit de la manière suivante :

$$\bar{T}_O = \sum_{i=1}^n \bar{T}_{F_i} . \quad (2.22)$$

□

Preuve Il suffit d'appliquer la définition 1 de la complexité moyenne en temps par rapport à l'ensemble de rayons issus de \mathcal{R} à l'expression de la complexité de la proposition 12. ■

Le corollaire précédent montre que la complexité moyenne en temps est linéairement dépendante du nombre de faces de l'objet considéré. Soit une scène composée de N objets BRep, chaque objet étant composé de n faces de même type, c'est-à-dire dont la complexité moyenne en temps de recherche de la première intersection avec un rayon est \bar{T}_F . En injectant l'expression 2.22 dans l'expression 2.20 après l'avoir moyennée, la complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et la scène géométrique s'écrit :

$$\bar{T}_{O^1, \dots, O^N} = n N \bar{T}_F .$$

2.4 Intersecteurs rayon-surface

Les modèles géométriques offrent différentes représentations (CSG et BRep en ce qui nous concerne) et surtout une grande diversité de représentations des surfaces qui sont la base des objets. En image de synthèse, et encore plus en optique radiométrique, la précision du calcul d'intersection entre un rayon et un objet et du calcul du vecteur normal en un point d'une surface est fondamentale. Cependant, cette précision ne doit pas être atteinte au détriment des performances des logiciels. Il est par conséquent nécessaire d'exploiter au mieux les propriétés particulières des surfaces utilisées. Ces surfaces sont connues par certains de leurs paramètres.

De manière simplifiée, les surfaces implicites sont du type :

$$F = \{x \in \mathbb{R}^3, f(x) = 0\} ,$$

où f est une fonction définie sur \mathbb{R}^3 à valeurs dans \mathbb{R} , connue à partir de la structure de données du modèle géométrique considéré. Dans le cas général, f est un polynôme à plusieurs variables et on connaît f par les coefficients de ce polynôme dans une base polynomiale. Dans le cas particulier qui nous intéresse, le plus courant dans le modèle géométrique CSG, où les formes élémentaires sont des sphères, ellipsoïdes, cylindres, cônes, f est un polynôme de degré 2 en les variables, la surface F est donc une quadrique connue par des paramètres géométriques (rayon, longueur, etc).

Les surfaces paramétriques sont du type :

$$F = \{x \in \mathbb{R}^3, x = g(s, t), (s, t) \in D \subset \mathbb{R}^2\} ,$$

où g est une fonction définie sur D à valeurs dans \mathbb{R}^3 , connue à partir de la structure de données du modèle géométrique considéré. Dans le cas général, g est une NURBS (*Non Uniform Rational Basis Spline*) ou une Bézier rationnelle. Dans le cas particulier qui nous intéresse, on prend en compte des symétries (axiales par exemple) ou des opérateurs constructifs (extrusion par exemple).

2.4.1 Surfaces implicites

Cas général

Une surface implicite F est le lieu des points x vérifiant des équations et/ou inéquations implicites décrites par un ensemble de paramètres :

$$F = \{x \in \mathbb{R}^3, f_0(x) = 0, f_i(x) \leq 0, i \in \{1, \dots, m\}\} ,$$

où $\{f_i\}_i$ sont des fonctions définies sur \mathbb{R}^3 à valeurs dans \mathbb{R} (figure 2.26). Les inéquations peuvent être prises au sens strict.

Pour calculer la première intersection entre un rayon $R_{x,w}$ et la surface F , il faut commencer par rechercher la première intersection entre $R_{x,w}$ et les surfaces décrites par les équations implicites $f_i(x) = 0$, pour $i \in \{0, \dots, m\}$. Ensuite, il ne reste plus qu'à retenir le point le plus proche de l'origine du rayon qui satisfait les inéquations implicites définissant la surface (figure 2.26). L'intersection entre $R_{x,w}$ et la surface d'équation implicite $f_i(x) = 0$ est un ensemble de points $\{x_{i,j}\}_j$ vérifiant :

$$\begin{cases} x_{i,j} \in R_{x,w} \\ f_i(x_{i,j}) = 0 \end{cases} .$$

Pour résoudre le système précédent, il suffit d'injecter l'équation du rayon dans f_i , et de rechercher les valeurs $\{t_{i,j}\}_j$ vérifiant :

$$f_i(x + t_{i,j} w) = 0, t_{i,j} > 0 .$$

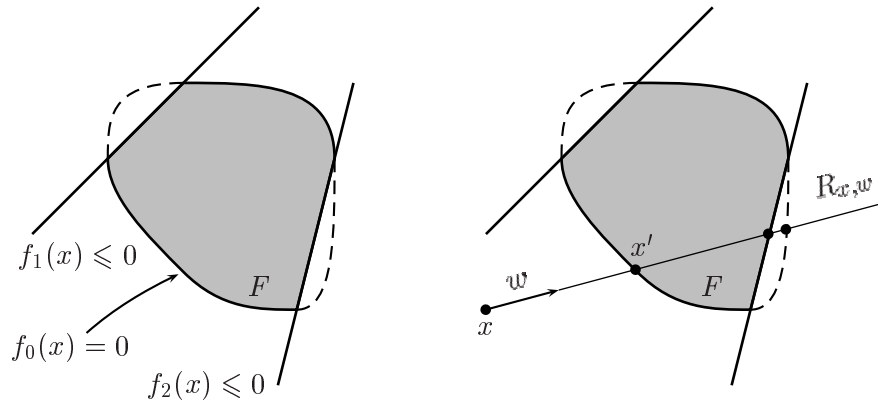


Fig. 2.26 – Représentation implicite et recherche de la première intersection avec un rayon.

La première intersection, si elle existe, est le point le plus proche de l'origine du rayon x qui satisfait les inéquations implicites :

$$\begin{cases} t' = \min_{i,j} \{t_{i,j}, f_i(x + t_{i,j} w) \leq 0, i \in \{1, \dots, m\}\} \\ x' = x + t' w \end{cases}$$

Les algorithmes de recherche de la première intersection se déroulent donc en deux étapes :

- substitution de l'équation du rayon dans l'expression de la surface considérée,
- recherche des zéros de la fonction univariée (le paramètre du rayon) ainsi obtenue.

La substitution est une opération numérique délicate. Le calcul symbolique est une alternative très intéressante. La programmation de l'algorithme de substitution n'est plus nécessaire, il suffit de coder directement les fonctions propre à chaque type de surface calculées symboliquement. Cependant, les logiciels de calcul symbolique ne donne pas toujours des résultats minimaux. Il est par conséquent important, en vue de la recherche des zéros, de simplifier le plus possible les expressions obtenues (Hanrahan [47]).

La recherche des zéros d'une fonction est un sujet vaste qui fait depuis longtemps l'objet de nombreux travaux. Aussi des études spécifiques ont été menées pour des fonctions particulières. Kalra et Barr [53] ont introduit une méthode robuste, basée sur la calcul des constantes de Lipschitz, adaptée aux surfaces algébriques généralisées introduites par Blinn [7]. Lorsque les fonctions considérées sont des polynômes la surface représentée est dite semi-algébrique, si de plus elle est décrite uniquement par l'équation $f_0(x) = 0$ (sans inéquation) on parle alors de surface algébrique. Ces surfaces particulières occupent une place importante en image de synthèse et prépondérante en optique radiométrique. Lorsque le degré du polynôme est inférieur ou égal à quatre la recherche des zéros est analytique. Par contre, pour des degrés plus élevés il faut utiliser une méthode numérique robuste. On distingue plusieurs types de méthodes qui permettent d'approcher les zéros d'un polynôme univarié. Ces méthodes incluent généralement une première phase de localisation ou de séparation des racines :

- la méthode de Collins-Akritas-Loos (Collins et Akritas [21], Collins et Loos [22]) qui permet de séparer les racines d'un polynôme qui ne possède pas de racine multiple,
- les méthodes géométriques utilisant les propriétés des polynômes de Bernstein (Farouki et Rajan [38] [39], Marchepoil [64]),
- les méthodes d'exclusion (Dedieu et Yakoubsohn [27], Marchepoil [64]) qui permettent d'éliminer les intervalles ne possédant pas de racine,
- les méthodes utilisant les séquences de Sturm et la règle des signes de Descartes (Durand [35], Hacques [46]),

Ensuite intervient une phase de calcul, accomplie par une méthode itérative utilisant une relation de récurrence (Hacquès [46]) telles que les méthodes de Newton-Raphson, Bairstow ou encore Legendre pour ne citer que celles-ci.

Une fois le point d'intersection déterminé, il est nécessaire de calculer le vecteur normal à la surface en ce point. Ce calcul se ramène à l'évaluation du gradient de la fonction décrivant la surface au point considéré (figure 2.27) :

$$u(x') = \nabla f_0(x') .$$

Dans le cas de surfaces implicites polynomiales, on peut utiliser un schéma de Hörner pour l'évaluation du gradient.

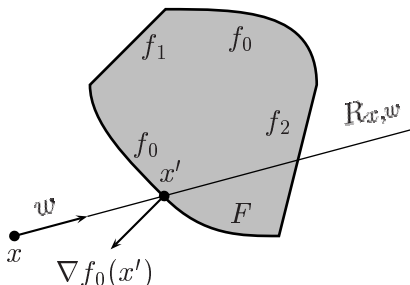


Fig. 2.27 – Calcul du vecteur normal.

Cas particulier

Dans le contexte qui nous intéresse, la plupart des surfaces sont au plus du second ordre, c'est-à-dire des quadriques. L'équation générale d'une quadrique s'écrit de la manière suivante (dans l'espace projectif) :

$$y^t Q y = 0 ,$$

où y^t est le vecteur transposé de y et Q est une matrice symétrique de dimension 4. Soit un rayon $R_{x,w}$ décrit par l'équation paramétrique suivante (dans l'espace projectif) :

$$y = x + t w , t > 0 .$$

La substitution de l'équation du rayon dans l'expression implicite de la quadrique donne une équation polynomiale univariée du second ordre :

$$a_0 + a_1 t + a_2 t^2 = 0 ,$$

où les coefficients $\{a_i\}_i$ sont donnés par :

$$\begin{cases} a_0 &= x^t Q x \\ a_1 &= 2 w^t Q x \\ a_2 &= w^t Q w \end{cases} .$$

Ainsi, la première intersection, si elle existe, est le point x' de paramètre t sur le rayon $R_{x,w}$ tel que t soit la plus petite solution strictement positive de l'équation de degré 2 précédente.

Les quadriques considérées dans le cadre de l'optique sont la sphère, l'ellipsoïde, le cylindre, le cône, la lentille et les surfaces polynomiales de degré inférieur à deux.

2.4.2 Surfaces paramétriques

Cas général

Une surface paramétrique F est décrite par une fonction $g = (g_1, g_2, g_3)$ définie sur un domaine $D \subset \mathbb{R}^2$ à valeurs dans \mathbb{R}^3 :

$$F = \{x \in \mathbb{R}^3, x = g(s, t), (s, t) \in D\} .$$

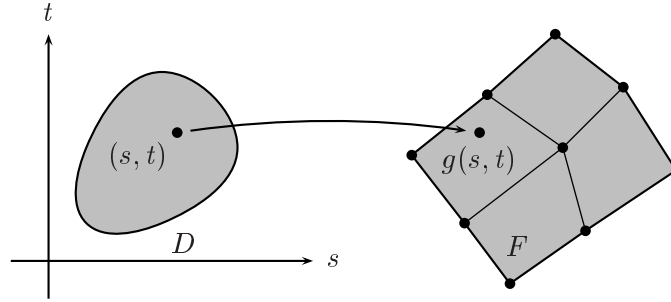


Fig. 2.28 – Représentation paramétrique.

Soit $R_{x,w}$ un rayon et $\{x_j\}_j$ l'ensemble des points d'intersection entre $R_{x,w}$ et F :

$$\begin{cases} x_j \in R_{x,w} \\ x_j \in F \end{cases} ,$$

ce qui signifie qu'il existe des valeurs du paramètre du rayon $\{t_j\}_j$ et des valeurs des paramètres de la surface $\{(s_j, t_j)\}_j$ telles que :

$$\begin{cases} x_j = x + t_j w, t_j > 0 \\ x_j = g(s_j, t_j), (s_j, t_j) \in D \end{cases} .$$

Selon les cas, on a besoin de connaître les valeurs $\{t_j\}_j$ du paramètre du rayon (test de visibilité ou calcul de distance), ou les valeurs $\{(s_j, t_j)\}_j$ des paramètres de la surface (calcul de vecteur normal à la surface), ou les points $\{x_j\}_j$ sur la surface, ou plusieurs de ces paramètres. Dans notre contexte, on s'intéresse au premier point d'intersection et au vecteur normal à la surface en ce point, on se contentera donc des $\{t_j\}_j$ et $\{(s_j, t_j)\}_j$. La première intersection entre $R_{x,w}$ et F , si elle existe, est définie par :

$$\begin{cases} t' = \min_j \{t_j\} \\ x' = x + t' w \end{cases} .$$

La méthode générale consiste à résoudre le système de trois équations à trois inconnues suivant :

$$x + t_j w - g(s_j, t_j) = 0 ,$$

et à ne retenir que les solutions satisfaisant $t_j > 0$ et $(s_j, t_j) \in D$.

Dans la majorité des systèmes CAO, les surfaces paramétriques sont des NURBS définies par produit tensoriel, éventuellement restreintes. Une NURBS définie par un produit tensoriel s'écrit sous la forme suivante :

$$F = \{x \in \mathbb{R}^3, x = g(s, t), (s, t) \in D\} ,$$

où D est un domaine rectangulaire :

$$D = [s_{\min}, s_{\max}] \times [t_{\min}, t_{\max}] .$$

On note alors $\{s_k\}_{k=0}^n$ et $\{t_\ell\}_{\ell=0}^m$ les points de D suivants :

$$\begin{cases} s_{\min} = s_0 \leq s_1 \leq \dots \leq s_{n-1} \leq s_n = s_{\max} \\ t_{\min} = t_0 \leq t_1 \leq \dots \leq t_{m-1} \leq t_m = t_{\max} \end{cases} ,$$

et on note $D_{k,\ell}$ le sous-domaine de D suivant (figure 2.29) :

$$D_{k,\ell} = [s_k, s_{k+1}] \times [t_\ell, t_{\ell+1}] .$$

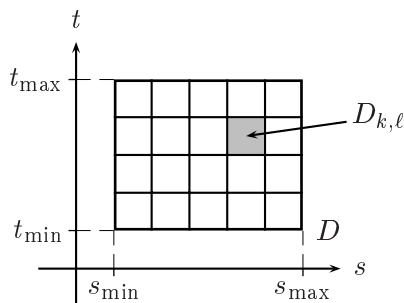


Fig. 2.29 – *Domaine des paramètres pour une NURBS naturelle.*

Soient $\{N_i^n\}_i$ et $\{N_j^m\}_j$ deux bases de splines polynomiales à une variable, respectivement par rapport à s et t . De plus, posons :

$$N_{i,j}^{n,m}(s, t) = N_i^n(s) N_j^m(t) , \quad \forall (s, t) \in D .$$

Une NURBS est alors définie par un ensemble de poids réels strictement positifs $\{\omega_{i,j}\}_{i,j}$, et un ensemble de points de \mathbb{R}^3 $\{a_{i,j}\}_{i,j}$, et on note :

$$g(s, t) = \frac{\sum_{i=0}^n \sum_{j=0}^m \omega_{i,j} a_{i,j} N_{i,j}^{n,m}(s, t)}{\sum_{i=0}^n \sum_{j=0}^m \omega_{i,j} N_{i,j}^{n,m}(s, t)} , \quad \forall (s, t) \in D ,$$

la NURBS correspondante. Sur chaque sous-domaine $D_{k,\ell}$, l'expression de $g(s, t)$ se réduit à une fonction rationnelle que l'on peut exprimer sous la forme :

$$g_{k,\ell}(s, t) = \frac{\sum_{i=0}^p \sum_{j=0}^q \omega_{i,j}^{k,\ell} a_{i,j}^{k,\ell} B_{i,j}^{p,q}(s, t)}{\sum_{i=0}^p \sum_{j=0}^q \omega_{i,j}^{k,\ell} B_{i,j}^{p,q}(s, t)} , \quad \forall (s, t) \in D_{k,\ell} .$$

Cette expression peut être obtenue par les algorithmes d'Oslo (Farin [37], Léon [62]), dont l'objectif est de fournir une méthode pour exprimer les $\{\omega_{i,j}^{k,\ell}\}_{i,j}$ et $\{a_{i,j}^{k,\ell}\}_{i,j}$ en fonction des $\{\omega_{i,j}\}_{i,j}$ et $\{a_{i,j}\}_{i,j}$. La base $\{B_{i,j}^{p,q}\}_{i,j}$ est souvent la base de Bernstein bivariee.

On remarque que dans le cas particulier où les poids $\omega_{i,j}$ sont tous égaux à 1, alors est ramené au cas polynomial (le dénominateur dans l'expression de $g(s, t)$ vaut 1).

Une NURBS restreinte est une NURBS définie sur un sous-ensemble de $[s_{\min}, s_{\max}] \times [t_{\min}, t_{\max}]$. La donnée d'une NURBS restreinte est :

- une NURBS, dite naturelle au sens de la définition précédente,
- une ou plusieurs courbes (NURBS éventuellement) fermées à valeurs dans $[s_{\min}, s_{\max}] \times [t_{\min}, t_{\max}]$ qui définissent le domaine D , généralement une courbe extérieure et une courbe intérieure (figure 2.30).

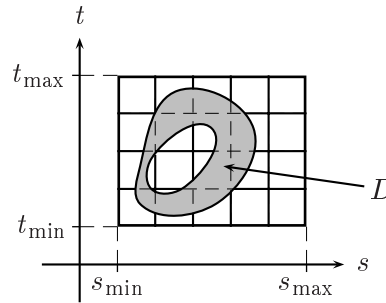


Fig. 2.30 – *Domaine des paramètres pour une NURBS restreinte.*

Les différents algorithmes doivent alors prendre en compte cette difficulté. En particulier, pour les calculs d'intersection il est nécessaire de vérifier l'appartenance au domaine D . Une autre difficulté est que, dans la pratique, les dessinateurs ou les concepteurs de systèmes mécaniques ou optiques, utilisent souvent des NURBS dégénérées (superposition de points de contrôle le plus souvent) qui créent de nombreux problèmes pour l'évaluation de la dérivée et particulièrement de la normale en un point de la surface. La bonne méthode serait de définir une NURBS restreinte.

Les différentes méthodes pour rechercher l'intersection entre un rayon et une surface paramétrique peuvent être classées en trois catégories :

- les méthodes de tessellation,
- les méthodes numériques itératives,
- les méthodes algébriques.

Les méthodes de tessellation consistent à approcher la surface par un ensemble de facettes. Les plus simples consistent à subdiviser la surface jusqu'à ce que chaque morceau considéré ait une courbure moyenne suffisamment faible pour être approché par un plan. Ces techniques de tessellation peuvent être combinées avec des techniques de localisation spatiale ayant pour but de diminuer le nombre de facettes à parcourir pour chaque rayon considéré. Whitted [86] utilise une hiérarchie de sphères englobantes pour subdiviser les surfaces. Sweeney et Bartels [82] subdivisent des surfaces B-splines et définissent une hiérarchie de volumes englobants en utilisant les propriétés particulières de la base de Bernstein.

Les méthodes numériques itératives (Toth [83] et Barr [5]) consistent à résoudre le problème d'intersection en exprimant, dans l'espace projectif, le rayon comme intersection de deux plans (Sweeney et Bartels [82]) :

$$\begin{cases} \sum_i a_i x_i = 0 \\ \sum_i b_i x_i = 0 \end{cases} .$$

Dans l'espace projectif, une surface paramétrique est représentée par une fonction $h = (h_1, h_2, h_3, h_4)$ définie sur un domaine $D \subset \mathbb{R}^2$ à valeurs dans \mathbb{R}^4 . En injectant les $\{h_i\}_i$ dans le système précédent, on

obtient un nouveau système de deux équations :

$$\begin{cases} \sum_i a_i h_i(s, t) = 0 \\ \sum_i b_i h_i(s, t) = 0 \end{cases} .$$

Pour résoudre ce système, une méthode numérique itérative, de type Newton, nécessite une phase de localisation qui consiste à définir une valeur initiale. C'est un problème très délicat. Certains cas particuliers donne lieu à des résolutions plus robustes. Pour des carreaux bi-cubiques chaque équation du système précédent est de degré six et en éliminant un des deux paramètres s ou t le système se réduit à une équation univariée de degré six (Kajiya [51]).

Les méthodes algébriques consistent à convertir les surfaces vers une représentation implicite. C'est la technique d'implicitisation, introduite par Sederberg [76]. On est alors ramené au cas de la recherche de la première intersection entre un rayon et une surface implicite. Steinberg [81] a présenté des résultats pour la conversion de carreaux bi-cubiques et bi-quadratiques. Sederberg et Anderson [75] ont traité les carreaux de Steiner (carreaux de Bézier bi-quadratique). Edwards [36] a développé des techniques pour convertir les superquadriques (produit sphérique de superconiques). Biard [6] présente une méthode qui traite les surfaces polynomiales de degré quelconque. Cependant, les techniques d'implicitisation sont difficiles à mettre en œuvre, l'augmentation de la complexité des fonctions à traiter étant le principal handicap. On les réservera donc à des cas très particuliers.

Une fois le point d'intersection déterminé, il est nécessaire de calculer le vecteur normal à la surface en ce point. De manière générale, le vecteur normal en un point $x = g(s, t)$ est donné par la relation suivante :

$$u(s, t) = \frac{\partial}{\partial s} g(s, t) \wedge \frac{\partial}{\partial t} g(s, t) .$$

Dans le cas de surfaces paramétriques polynomiales, on peut utiliser un schéma de Hörner pour l'évaluation des dérivées partielles.

Cas particulier

Dans le contexte qui nous intéresse, les surfaces considérées sont souvent de degré 2 en chaque variable, ce sont donc des quadriques, ce qui nous ramène au cas implicite. Pour les surfaces dites semi-paramétriques, telles que les surfaces de révolution ou les surfaces extrudées, on utilise des méthodes spécifiques (Dessarce [32] pour les surfaces de révolution).

2.5 Conclusion

Ce chapitre donne une description précise de la représentation des objets pour les deux modèles géométriques considérés ainsi que les algorithmes de base nécessaires à la recherche d'intersections entre des rayons et des objets. On retiendra, en particulier, que les spécificités du domaine de l'optique posent des problèmes quant au choix d'un modèle géométrique. On est toujours partagé entre le modèle CSG et le modèle BRep. Pour le premier, les propriétés optiques sont naturellement représentées pour les surfaces et les volumes et les algorithmes d'intersection sont très rapides pour des objets primitifs de type implicite. Pour le second, la représentation des propriétés optiques pour les volumes peut poser des problèmes et les algorithmes d'intersection sont plus coûteux que dans le cas CSG. Cependant, le modèle BRep s'impose comme un standard au niveau industriel, notamment pour la représentation des données géométriques, d'où sa nécessité.

L'étude des complexités des algorithmes d'intersection a pour but de donner une idée précise des coûts en temps à prendre en compte pour la suite. Notamment pour le cas CSG, l'étude comparative entre

la méthode des epsilons et la méthode de Roth montre que pour ce type d'objet il est crucial de choisir une méthode qui limite les parcours d'arbre.

Dans la suite, on supposera disposer de ces deux modèles géométriques et des algorithmes d'intersection associés.

Chapitre 3

Localisation spatiale pour le calcul d'intersection

Les différentes méthodes de résolution envisagées sont susceptibles de faire intervenir les notions de rayon lumineux, de visibilité, en tout cas d'intersection entre un rayon et un objet. On peut considérer actuellement un rapprochement entre études radiométriques et synthèse d'image. Dans le premier cas, on cherche à calculer au plus près le phénomène physique fondé sur les équations étudiées au premier chapitre. Dans le second cas, on est d'avantage intéressé par le résultat visuel. Cependant, les modèles géométriques sont comparables dans les deux domaines, même si les géométries ne sont pas soumises aux mêmes contraintes de précision dans les deux cas.

Le problème principal est l'amélioration du temps de simulation en accélérant les calculs d'intersection entre les rayons et les objets. Dans le premier paragraphe on présentera de façon succincte les méthodes d'accélération traditionnelles pour les calculs d'intersection. Dans le second paragraphe on détaillera la méthode envisagée pour notre contexte. Enfin, dans un troisième paragraphe on présentera des résultats expérimentaux dans l'environnement industriel.

3.1 Accélération des calculs d'intersection

3.1.1 Où gagner du temps ?

Dans une simulation, la majeure partie du temps de calcul est consacrée à la recherche de la première intersection entre des rayons et le système considéré. Whitted [86] a évalué à au moins 95% le temps passé à effectuer de tels calculs sur des scènes complexes. Bien que les modèles physiques utilisés dans nos applications soient plus complexes que ceux généralement utilisés en image de synthèse (notamment pour la modélisation du noyau de dispersion surfacique¹), l'expérience montre qu'au moins 80% du temps est consacré aux calculs d'intersection. D'où la nécessité d'employer des méthodes d'accélération des calculs d'intersection adaptées et efficaces, dont les plus courantes sont présentées dans la suite.

La gain de temps sur une simulation passe donc par une accélération des calculs d'intersection. Si l'on regarde de plus près, on se rend compte que l'algorithme de recherche de la première intersection entre un rayon et une scène géométrique O^1, \dots, O^N comporte deux parties (algorithme 6). La première partie consiste à parcourir tous les objets de la scène considérée. La seconde partie concerne les calculs d'intersection rayon-objet. Bien entendu, il s'agit ici d'une méthode naïve. Cependant son importance réside dans le fait que son analyse permet de classer les méthodes d'accélération des calculs d'intersection.

¹Voir définition au paragraphe 1.2.5.

Algorithme 6: *Méthode naïve.*

```

INTERRAYONSCENE-NAIF( $x, w, O^1, \dots, O^N$ )
(1)   ( $i_{\min}, t_{\min}, I_{\min}$ )  $\leftarrow$  ( $-1, +\infty, \{\}$ )
(2)   for  $i = 1$  to  $N$ 
(3)     ( $t, I$ )  $\leftarrow$  INTERRAYONOBJET( $x, w, O^i$ )
(4)     if  $t < t_{\min}$ 
(5)       ( $i_{\min}, t_{\min}, I_{\min}$ )  $\leftarrow$  ( $i, t, I$ )
(6)   return ( $i_{\min}, t_{\min}, I_{\min}$ )

```

L'algorithme INTERRAYONOBJET recherche la première intersection entre un rayon et un objet. Si une intersection existe, alors il renvoie le paramètre correspondant sur le rayon et, éventuellement, diverses informations relatives à l'objet et au point d'intersection trouvé (vecteur normal au point trouvé, point dans l'espace des paramètres pour un objet paramétrique, numéro de la face pour un objet BRep, etc). Sinon, par convention, il renvoie $(+\infty, \{\})$. La proposition suivante donne une expression de la complexité en temps relative à l'algorithme 6.

Proposition 13

Soit $R_{x,w}$ un rayon. Soient O^1, \dots, O^N des objets. La complexité moyenne en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et la scène géométrique O^1, \dots, O^N s'écrit de la manière suivante :

$$T_{O^1, \dots, O^N}(x, w) = \sum_{i=1}^N T_{O^i}(x, w), \quad (3.1)$$

où $T_{O^i}(x, w)$ est la complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et l'objet O^i . □

Preuve L'expression de la complexité en temps est évidente à la lecture de l'algorithme 6. ■

Le corollaire suivant donne la complexité moyenne en temps par rapport à un ensemble de rayons issus de \mathcal{R} (voir définition page 32).

Corollaire 4

Soient O^1, \dots, O^N des objets. La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et la scène géométrique O^1, \dots, O^N s'écrit de la manière suivante :

$$\bar{T}_{O^1, \dots, O^N} = \sum_{i=1}^N \bar{T}_{O^i}, \quad (3.2)$$

où \bar{T}_{O^i} est la complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et l'objet O^i . □

Preuve Il suffit d'appliquer la définition 1 page 36 de la complexité moyenne en temps à l'expression 3.1. ■

Accélérer les calculs d'intersection c'est réduire la complexité $\bar{T}_{O^1, \dots, O^N}$. Considérant que la précision numérique des calculs d'intersection est fixée, les méthodes d'accélération peuvent être classées en deux familles :

- l'accélération des calculs d'intersection rayon-objet, c'est-à-dire la réduction du coût de l'algorithme INTERRAYONOBJET, donc la diminution des complexités moyennes en temps \overline{T}_{O_i} ,
- l'accélération des calculs d'intersection rayon-scène, c'est-à-dire la diminution de taille de la boucle dans l'algorithme 6, donc la diminution de la taille de la liste des objets à parcourir, caractérisée par la somme dans l'équation 3.2.

Dans la première famille, on distingue deux types d'accélération. D'une part, l'amélioration des méthodes numériques présentées dans le paragraphe 2.4, qui consistent essentiellement en la mise en œuvre de techniques robustes et rapides de recherche des zéros d'équations algébriques. Ces techniques dépendent du type de surface considérée. D'autre part, la mise en place de méthodes géométriques d'exclusion dont le but est de réduire les coûts pour des rayons éloignés. Ces méthodes utilisent les propriétés des surfaces ou des modèles géométriques considérés. Elles sont présentées dans le paragraphe 3.1.2.

La seconde famille correspond aux méthodes de partition de la liste des objets considérés. Elles consistent en la définition d'une subdivision de l'espace contenant la scène géométrique. Le paragraphe 3.1.3 décrit le principe de ces méthodes.

La mise en œuvre d'une méthode dépend de plusieurs paramètres. Les contraintes sont d'autant plus nombreuses que la mise en place d'une méthode d'accélération doit se faire dans un contexte industriel. Le paragraphe 3.1.4 justifie le choix d'une méthode en fonction du cahier des charges.

3.1.2 Utilisation d'objets englobants

Le principe des méthodes à objet englobant est d'associer à chaque objet O un objet englobant O_e (figure 3.1) qui permette des tests d'intersection très rapides avec un rayon :

$$\left\{ \begin{array}{l} O \subset O_e \\ T_{O_e}(x, w) \ll T_O(x, w), \forall (x, w) \in \mathcal{R} \end{array} \right. ,$$

où T_O et T_{O_e} sont les complexités en temps respectivement pour les objets O et O_e .

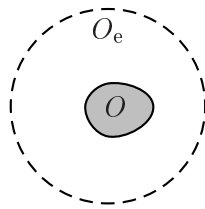


Fig. 3.1 – Définition d'un objet englobant.

L'intérêt de ces méthodes est d'éviter le calcul d'intersection avec O pour les rayons qui n'ont pas d'intersection avec l'objet englobant associé. Ce sont des méthodes d'exclusion dans le sens où elles consistent à ne pas tenir compte des rayons dont on est sûr qu'ils n'ont pas d'intersection avec l'objet O . Le critère d'exclusion géométrique est caractérisée par (figure 3.2) :

$$R_{x,w} \cap O_e = \emptyset \implies R_{x,w} \cap O = \emptyset .$$

L'algorithme de recherche de la première intersection entre un rayon et un objet consiste alors à tester systématiquement l'intersection avec l'objet englobant puis s'il y a intersection à tester l'intersection avec l'objet considéré (algorithme 7).

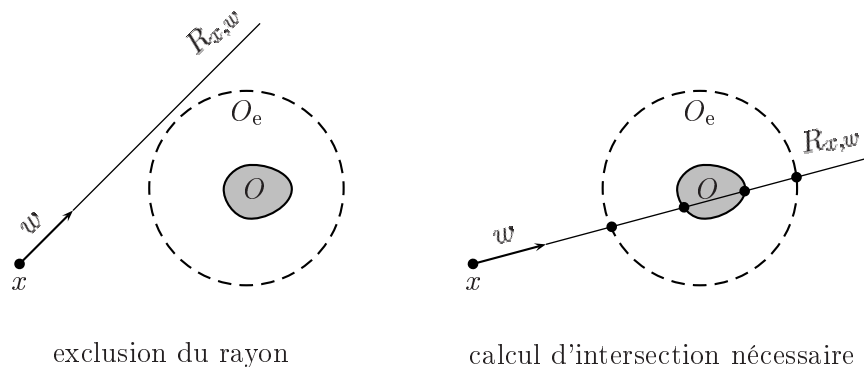


Fig. 3.2 – Propriété d'exclusion pour un objet englobant.

Algorithme 7: *Objet englobant.*INTERRAYONOBJET-ENGLLOBANT(x, w, O)

- (1) $(t_{\min}, I) \leftarrow \text{INTERRAYONOBJET}(x, w, O_e)$
- (2) $I_{\min} \leftarrow \{\}$
- (3) **if** $t_{\min} < +\infty$ **then** $(t_{\min}, I_{\min}) \leftarrow \text{INTERRAYONOBJET}(x, w, O)$
- (4) **return** (t_{\min}, I_{\min})

L'algorithme INTERRAYONOBJET recherche la première intersection entre un rayon et un objet. Si une intersection existe, alors il renvoie le paramètre correspondant sur le rayon et, éventuellement, diverses informations relatives à l'objet et au point d'intersection trouvé (vecteur normal au point trouvé, point dans l'espace des paramètres pour un objet paramétrique, numéro de la face pour un objet BRep, etc). Sinon, par convention, il renvoie $(+\infty, \{\})$.

Pour estimer la complexité en temps de l'algorithme précédent, on a besoin de la fonction indicatrice d'intersection avec un objet englobant (définition 2 page 55). On rappelle que $\mathbb{1}_{\mathcal{R}, O_e}$ une fonction définie sur \mathcal{R} à valeurs dans $\{0, 1\}$, dont l'expression est la suivante :

$$\mathbb{1}_{\mathcal{R}, O_e}(x, w) = \begin{cases} 1 & \text{si } R_{x,w} \cap O_e \neq \emptyset \\ 0 & \text{sinon} \end{cases} .$$

La proposition suivante donne une expression de la complexité en temps relative à l'algorithme 7.

Proposition 14

Soit $R_{x,w}$ un rayon. Soit O un objet. Soit O_e un objet englobant O , c'est-à-dire tel que $O \subset O_e$. La complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et l'objet O s'écrit de la manière suivante :

$$T_{O, O_e}(x, w) = T_{O_e}(x, w) + \mathbb{1}_{O_e}(x, w) T_O(x, w) , \quad (3.3)$$

où $T_{O_e}(x, w)$ et $T_O(x, w)$ sont les complexités en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et respectivement les objets O_e et O .

□

Preuve Elle est évidente à la lecture de l'algorithme 7.

■

De même que pour l'indicatrice d'intersection, on a besoin de la caractérisation de la proportion de rayons qui ont une intersection avec un objet englobant (définition 3 page 59). On rappelle que la

proportion de rayons issus de \mathcal{R} qui ont une intersection avec un objet O_e est donnée par le nombre réel suivant :

$$\gamma_{\mathcal{R},O_e} = \frac{1}{\text{mes}(\mathcal{R})} \int_{\mathcal{R}} \mathbb{1}_{O_e}(x, w) d\mu(x) d\sigma(w) .$$

Pour différencier les rayons qui ont une intersection avec un objet englobant O_e des autres, on introduit les deux sous-ensembles suivants :

$$\begin{cases} \mathcal{R}_{\text{in},O_e} = \{(x, w) \in \mathcal{R}, R_{x,w} \cap O_e \neq \emptyset\} \\ \mathcal{R}_{\text{ex},O_e} = \mathcal{R} \setminus \mathcal{R}_{\text{in},O_e} \end{cases} .$$

$\mathcal{R}_{\text{in},O_e}$ est donc l'ensemble des rayons issus de \mathcal{R} qui ont une intersection avec O_e , $\mathcal{R}_{\text{ex},O_e}$ est l'ensemble complémentaire dans \mathcal{R} .

Le corollaire suivant donne la complexité moyenne en temps relative à l'algorithme 7.

Corollaire 5

Soit O un objet. Soit O_e un objet englobant O , c'est-à-dire tel que $O \subset O_e$. Soit $\gamma_{\mathcal{R},O_e}$ la proportion de rayons issus de \mathcal{R} qui ont une intersection avec O_e . La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et l'objet O s'écrit de la manière suivante :

$$\bar{T}_{O,O_e} = \bar{T}_{O_e} + \gamma_{\mathcal{R},O_e} \bar{T}_O^{\text{in}} , \quad (3.4)$$

où \bar{T}_{O_e} est la complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et l'objet O_e et \bar{T}_O^{in} est la complexité moyenne en temps de recherche de la première intersection entre un rayon issu de $\mathcal{R}_{\text{in},O_e}$ et l'objet O . □

Preuve Il suffit d'appliquer la définition 1 page 36 de la complexité moyenne en temps à l'expression 3.3. Le premier terme ne pose aucun problème. Regardons de plus près le second. En remarquant que l'indicatrice d'intersection sur O_e relativement à \mathcal{R} est nulle sur $\mathcal{R}_{\text{ex},O_e}$, la moyenne du second terme de l'expression 3.3 s'écrit de la manière suivante :

$$\frac{1}{\text{mes}(\mathcal{R})} \int_{\mathcal{R}} \mathbb{1}_{O_e}(x, w) T_O(x, w) d\mu(x) d\sigma(w) = \frac{1}{\text{mes}(\mathcal{R})} \int_{\mathcal{R}_{\text{in},O_e}} T_O(x, w) d\mu(x) d\sigma(w) .$$

Donc l'expression précédente devient :

$$\frac{1}{\text{mes}(\mathcal{R})} \int_{\mathcal{R}} \mathbb{1}_{O_e}(x, w) T_O(x, w) d\mu(x) d\sigma(w) = \frac{\text{mes}(\mathcal{R}_{\text{in},O_e})}{\text{mes}(\mathcal{R})} \bar{T}_O^{\text{in}} .$$

Il ne reste plus qu'à remarquer que le rapport des deux mesures est la proportion de rayons issus de \mathcal{R} qui ont une intersection avec O_e , c'est-à-dire le nombre $\gamma_{\mathcal{R},O_e}$. D'où l'expression proposée. ■

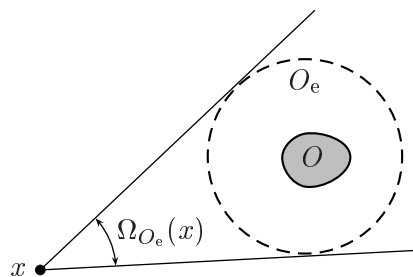
L'expression 3.4 rend compte de l'efficacité de la méthode à objet englobant. En effet, l'efficacité augmente lorsque le paramètre $\gamma_{\mathcal{R},O_e}$ et la complexité moyenne en temps \bar{T}_{O_e} diminuent. En d'autres termes, cela signifie que moins l'objet englobant est traversé par des rayons et plus sa forme est simple, plus la méthode est efficace. Il faut donc tenter d'utiliser des englobants petits dans le système considéré et de formes simples. C'est un compromis entre la dimension de l'englobant et sa simplicité. Différents types d'englobants ont été testés.

- Les sphères, introduites par Whitted [86], sont l'objet englobant le plus simple qui puisse exister, mais sa dimension relative à l'objet englobé n'est pas contrôlable.

- Les ellipsoïdes, introduits par Bouville [9], permettent de mieux contrôler la dimension des englobants, mais le choix du meilleur ellipsoïde est difficile et les temps de calculs d'intersection sont beaucoup plus longs que pour la sphère.
- Les cylindres, introduits par Weghorst, Hooper et Greenberg [84], permettent de privilégier une orientation, mais les temps de calculs d'intersection sont longs.
- Les boîtes englobantes, introduites par Rubin et Whitted [70], sont des parallélépipèdes. Le choix de leur orientation permet de minimiser leur taille. De plus, les calculs d'intersection sont rapides. Roth [69] utilise des boîtes englobantes parallèles aux axes pour des objets de type CSG. Le principal inconvénient est que la dimension relative à l'objet englobé n'est pas contrôlable.
- Les polyèdres, introduits par Kay et Kajiya [55] comme une généralisation des boîtes englobantes mais avec un nombre de directions limitées, généralisés par Roelens [68] qui admet des directions quelconques, permettent d'approcher encore mieux les objets.

Exemple 8

Soit O un objet. Soit O_e une sphère de rayon r englobant O . On suppose que la complexité en temps relative à l'objet O vérifie les hypothèses du corollaire 5.



La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et l'objet O par la méthode à objet englobant est donnée par l'expression 3.4 :

$$\bar{T}_{O,O_e} = \bar{T}_{O_e} + \gamma_{\mathcal{R},O_e} T_O .$$

Soit x un point. Considérons l'ensemble de rayons issus de $\mathcal{R} = \{x\} \times \mathcal{S}^2$. On suppose donc que les rayons sont issus d'une source ponctuelle dont la propriété d'émission est uniforme dans toutes les directions de l'espace. La proportion de rayons issus de \mathcal{R} qui ont une intersection avec O_e est le rapport de la mesure de l'angle solide issu de x interceptant O_e et de la mesure de la sphère unité :

$$\gamma_{\mathcal{R},O_e} = \frac{\text{mes}(\Omega_{O_e}(x))}{\text{mes}(\mathcal{S}^2)} = \frac{\text{mes}(\Omega_{O_e}(x))}{4\pi} .$$

La mesure de l'angle solide $\Omega_{O_e}(x)$ dépend de la position de x . Si x est dans O_e alors la mesure est celle de la sphère unité. Sinon, c'est l'aire de la calotte de sphère de centre x et de rayon $\sqrt{d^2 - r^2}$, où d est la distance entre x et le centre de la sphère O_e , divisée par le carré du rayon de cette sphère :

$$\text{mes}(\Omega_{O_e}(x)) = \begin{cases} 4\pi & \text{si } x \in O_e \\ 2\pi \left(1 - \sqrt{1 - \left(\frac{r}{d}\right)^2} \right) & \text{si } x \notin O_e \end{cases} .$$

La complexité moyenne en temps \bar{T}_{O_e} peut s'écrire sous la forme :

$$\bar{T}_{O_e} = \alpha T_O ,$$

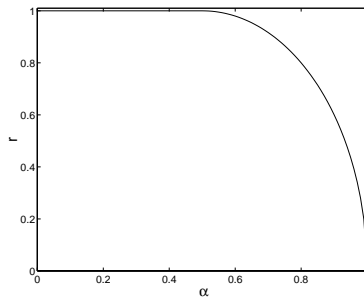
où α est un paramètre dans $[0, 1]$ (O_e est par définition plus simple que O). La complexité moyenne en temps pour la méthode à objet englobant s'écrit donc de la manière suivante :

$$\bar{T}_{O,O_e} = \begin{cases} (\alpha + 1)T_O & \text{si } x \in O_e \\ \left(\alpha + \frac{1}{2} \left(1 - \sqrt{1 - \left(\frac{r}{d}\right)^2} \right) \right) T_O & \text{si } x \notin O_e \end{cases} .$$

Cette expression montre que pour un point $x \in O_e$ la méthode n'est pas efficace, ce qui est évident puisque le critère d'exclusion géométrique n'est jamais vérifié. Pour un point $x \notin O_e$, l'efficacité dépend des paramètres α , r et d . La distance d est arbitraire, on peut prendre $d = 1$ et faire varier les deux autres paramètres dans $[0, 1]$ (r est nécessairement plus petit que 1 puisque x n'est pas dans O_e). Ainsi, un couple (α, r) est caractéristique de l'objet englobant O_e . α caractérise sa simplicité relativement à O , tandis que r caractérise sa taille relativement à la distance au point x (car on a fixé d). L'efficacité de la méthode à englobant est donnée par la valeur du facteur devant T_O dans l'expression précédente. Celui-ci est inférieur à 1 pour les couples (α, r) tels que :

$$(\alpha, r) \in \left(\left[0, \frac{1}{2} \right] \times [0, 1] \right) \cup \left(\left[\frac{1}{2}, 1 \right] \times \{ r, r \leq 2d\sqrt{(1-\alpha)\alpha} \} \right) .$$

Le graphe suivant illustre l'ensemble des couples admissibles. La zone en dessous de la courbe représente donc les couples (α, r) pour lesquels la méthode à objet englobant est plus efficace que la méthode classique. On remarque que pour $\alpha \leq 1/2$ la méthode à objet englobant est toujours meilleure quel que soit r . En d'autres termes, dès que la simplicité de l'objet O_e par rapport à l'objet O est telle que la complexité \bar{T}_{O_e} soit au moins deux fois plus petite que la complexité T_O , alors la méthode est efficace quel que soit l'origine des rayons considérée en dehors de l'objet O_e .



□

3.1.3 Localisation spatiale

Les méthodes de localisation spatiale sont basées sur la propriété de cohérence spatiale entre les objets et les rayons. Le but est de réduire le nombre d'intersections à calculer pour chaque rayon considéré. Le principe est d'effectuer une partition de l'espace contenant les objets de la scène géométrique, puis de rechercher la première région traversée par le rayon et qui contient des objets. Si une intersection est trouvée alors on s'arrête, sinon on progresse dans la partition. Ces méthodes comportent donc deux phases :

- la phase de pré-calcul consistant à définir une partition des objets,
- la phase de calcul consistant à parcourir la partition.

La performance globale des méthodes de localisation dépend de deux paramètres : le pouvoir séparant de la partition et la facilité de suivi du rayon. Les partitions non régulières permettent une meilleure précision alors que les partitions régulières facilitent le suivi du rayon. La définition suivante caractérise de manière générale une partition d'un ensemble d'objets.

Définition 4

Soient O^1, \dots, O^N des objets. Une partition de O^1, \dots, O^N est un ensemble $\{R_k\}_{k=1}^n$ de régions de \mathbb{R}^3 tel que :

$$\forall i \in \{1, \dots, N\}, O^i \subset \bigcup_{k=1}^n R_k .$$

□

La définition suivante caractérise l'ensemble des points d'un objet qui interagissent avec des rayons (on ne s'intéresse qu'aux intersections rayon-surface).

Définition 5

Soit O un objet. On appelle surface optique de O , et on note $\Sigma(O)$, l'objet suivant :

$$\Sigma(O) = \begin{cases} O & \text{si } O \text{ est surfacique} \\ \partial O & \text{si } O \text{ est volumique} \end{cases} ,$$

où ∂O représente le contour de O .

□

D'un point de vue pratique, les régions issues d'une partition sont soit des objets englobants (des sphères, des ellipsoïdes, des cylindres, des boîtes englobantes ou des parallélépipèdes), soit des polyèdres (définis par intersections de demi-espaces), soit des voxels (parallélépipèdes parallèles aux axes). Localiser les objets O^1, \dots, O^N dans une partition $\{R_k\}_{k=1}^n$, c'est associer à chaque région R_k des informations décrivant les propriétés géométriques et physiques propres à cette région et donc aux objets qui s'y trouvent. Il est au moins nécessaire d'associer la liste des objets ayant une intersection avec la région considérée, c'est-à-dire de localiser les objets dans la partition.

Définition 6

Soient O^1, \dots, O^N des objets. Soit $\{R_k\}_{k=1}^n$ une partition de ces objets. On note L_{R_k} la liste des indices des objets qui possèdent une intersection avec la région R_k :

$$L_{R_k} = \{i \in \{1, \dots, N\} \text{ tel que } \Upsilon(O^i) \cap R_k \neq \emptyset\} ,$$

où $\Upsilon(O^i)$ est l'objet considéré pour l'évaluation de l'intersection avec R_k :

$$\Sigma(O^i) \subset \Upsilon(O^i) .$$

□

En pratique, pour un objet O donné, $\Upsilon(O)$ est soit la surface optique, c'est-à-dire $\Sigma(O)$, soit un objet englobant l'objet O .

Une fois la partition de la scène géométrique mise en place, il reste à définir le parcours du rayon. Quelle que soit la méthode de localisation spatiale considérée, la partition de l'ensemble des objets est associée à une structure de données qui peut être vue comme un arbre, où chaque nœud est lié à une région de la partition, et, par conséquent, à une liste d'objets. Fort de ce constat, les algorithmes de parcours du rayon dans une partition peuvent être classés en deux catégories :

- les algorithmes itératifs (algorithme 8),
- les algorithmes récursifs (algorithme 9).

Algorithme 8: *Localisation spatiale (parcours itératif).*

```

INTERRAYONSCENE-ITERATIF( $x, w, arbre$ )
(1)  ( $i_{\min}, t_{\min}, I_{\min}$ )  $\leftarrow (-1, +\infty, \{\})$ 
(2)   $fin \leftarrow \mathbf{false}$ 
(3)   $k \leftarrow -1$ 
(4)  while not  $fin$ 
(5)     $k \leftarrow \text{PARCOURSRAYONPARTITION}(x, w, arbre, k)$ 
(6)    if  $k \neq -1$ 
(7)      for  $j = 1$  to  $\text{taille}(L_{R_k})$ 
(8)         $i \leftarrow L_{R_k}(j)$ 
(9)         $(t, I) \leftarrow \text{INTERRAYONOBJET}(x, w, O^i)$ 
(10)       if  $t < t_{\min}$  then  $(i_{\min}, t_{\min}, I_{\min}) \leftarrow (i, t, I)$ 
(11)       if  $t_{\min} < +\infty$  then  $fin \leftarrow \mathbf{true}$ 
(12)     else
(13)        $fin \leftarrow \mathbf{true}$ 
(14)  return  $(i_{\min}, t_{\min}, I_{\min})$ 

```

La variable $arbre$ représente la structure de données qui lie les régions $\{R_k\}_{k=1}^n$ aux objets O^1, \dots, O^N . L'algorithme PARCOURSRAYONPARTITION renvoie l'indice de la première région traversée par un rayon et qui contient des objets. Par convention, il renvoie -1 si aucune région n'est traversée. Pour initialiser le parcours, cet algorithme prend en paramètre d'entrée l'indice de la dernière région rencontrée. L'algorithme INTERRAYONOBJET recherche la première intersection entre un rayon et un objet. Si une intersection existe, alors il renvoie le paramètre correspondant sur le rayon et, éventuellement, diverses informations relatives à l'objet et au point d'intersection trouvé (vecteur normal au point trouvé, point dans l'espace des paramètres pour un objet paramétrique, numéro de la face pour un objet BRep, etc). Sinon, par convention, il renvoie $(+\infty, \{\})$.

Algorithme 9: *Localisation spatiale (parcours récursif).*

```

INTERRAYONSCENE-RECURSIF( $x, w, n\aeud$ )
(1)  ( $i_{\min}, t_{\min}, I_{\min}$ )  $\leftarrow (-1, +\infty, \{\})$ 
(2)   $k \leftarrow \text{REGION}(n\aeud)$ 
(3)  if  $L_{R_k} \neq \emptyset$ 
(4)    for  $j = 1$  to  $\text{taille}(L_{R_k})$ 
(5)       $i \leftarrow L_{R_k}(j)$ 
(6)       $(t, I) \leftarrow \text{INTERRAYONOBJET}(x, w, O^i)$ 
(7)      if  $t < t_{\min}$  then  $(i_{\min}, t_{\min}, I_{\min}) \leftarrow (i, t, I)$ 
(8)   $n\aeud\text{-courant} \leftarrow \text{INITPARCOURSRAYONARBRE}(n\aeud)$ 
(9)  while  $n\aeud\text{-courant} \neq \mathit{nil}$ 
(10)    $(i, t, I) \leftarrow \text{INTERRAYONSCENE-RECURSIF}(x, w, n\aeud\text{-courant})$ 
(11)   if  $t < t_{\min}$  then  $(i_{\min}, t_{\min}, I_{\min}) \leftarrow (i, t, I)$ 
(12)    $n\aeud\text{-courant} \leftarrow \text{PARCOURSRAYONARBRE}(x, w, n\aeud\text{-courant})$ 
(13)  return  $(i_{\min}, t_{\min}, I_{\min})$ 

```

La variable $n\aeud$ représente un nœud de l'arbre représentant la structure de données qui lie les régions $\{R_k\}_{k=1}^n$ aux objets O^1, \dots, O^N . L'algorithme REGION est une fonction d'accès à la région associée à un nœud donné. L'algorithme INTERRAYONOBJET recherche la première intersection entre un rayon et un objet. Si une intersection existe, alors il renvoie le paramètre correspondant sur le rayon et, éventuellement, diverses informations relatives à l'objet et au point d'intersection trouvé (vecteur normal au point trouvé, point dans l'espace des paramètres pour un objet paramétrique, numéro de la face pour

un objet BRep, etc). Sinon, par convention, il renvoie $(+\infty, \{\})$. L'algorithme `INITPARCOURSRAYONARBRE` initialise le parcours de l'arbre. Il renvoie le nœud correspondant au premier fils rencontré, dans un sens de parcours défini par la méthode considérée. Si aucun nœud n'est pas trouvé, alors, par convention, il renvoie *nil*. L'algorithme `PARCOURSRAYONARBRE` recherche dans l'arbre la prochaine région traversée par un rayon, dans un sens de parcours défini par la méthode considérée. Si une région existe alors il renvoie le nœud correspondant. Sinon, par convention, il renvoie *nil*.

La définition 4 est très générale en ce sens qu'elle n'impose aucune restriction sur le nombre de régions ainsi que sur leur type. Ainsi, le domaine contenant l'ensemble des objets ou encore l'ensemble des objets englobants sont bien des partitions. Cependant, l'intérêt d'une partition est bien évidemment de considérer plusieurs régions. Il existe différentes méthodes pour définir une partition. Relativement aux notions d'objet et de scène géométrique, on distingue trois classes :

- les méthodes locales,
- les méthodes globales,
- les méthodes mixtes.

Les algorithmes 8 et 9 sont écrits dans un contexte général, chaque méthode présentée dans la suite est rattachée à une variante, voire une utilisation conjointe de ces deux algorithmes.

Méthodes locales

Ces méthodes consistent à partir des objets pour construire une partition de la scène géométrique. Leur principe est d'utiliser les objets englobants des objets de la scène géométrique comme régions de base pour la construction de la partition. Différents types de partition ont été développés :

- hiérarchie d'objets englobants,
- spécifique aux objets CSG.

Le principe des méthodes de partition par hiérarchie d'objets englobants est d'inclure les objets dans des objets englobants, puis de regrouper ces objets englobants dans d'autres objets englobants. La hiérarchisation consiste à répéter cette étape (figure 3.3). Les régions d'une partition sont donc des objets englobants et définir une partition c'est regrouper les régions dans d'autres régions jusqu'à obtenir une région qui englobe toute la scène géométrique. Par nature de ces méthodes, les structures de données liées à ce type de partition sont arborescentes. Chaque nœud est associé un objet englobant. Une feuille est associée à un seul objet de la scène géométrique. Un nœud autre qu'une feuille n'est associé à aucun objet de la scène géométrique. La racine de l'arbre est donc associée à l'objet englobant la scène géométrique. La localisation des objets dans les différentes régions est induite par la construction de la structure de données et ne pose donc aucun problème (on n'utilise pas la définition 6 pour construire les listes de liens).

Le parcours du rayon consiste à rechercher les intersections avec les objets englobants en partant de la racine de l'arbre. Pour chaque objet englobant, on calcule l'intersection avec le rayon considéré. S'il y a intersection, alors on recherche l'intersection entre le rayon considéré et les objets englobés. Et ainsi de suite, jusqu'à rencontrer une feuille de l'arbre, c'est-à-dire un objet de la scène géométrique. Naturellement ce processus induit un algorithme de type récursif.

Rubin et Whitted [70] furent les premiers à utiliser une telle méthode de partition. Les objets englobants sont des boîtes englobantes, non nécessairement parallèles aux axes de manière à minimiser leur taille. La difficulté de ces méthodes de partition est de choisir les meilleurs objets englobants et d'organiser la structure de données de manière à minimiser la complexité en temps de la recherche d'intersection. Différents auteurs ont travaillé à l'amélioration de la méthode initiale.

Weghorst, Hooper et Greenberg [84] utilisent une expression de la complexité en temps de calcul de l'intersection entre un rayon et un objet englobant, semblable à celle donnée par le corollaire 5, dans le

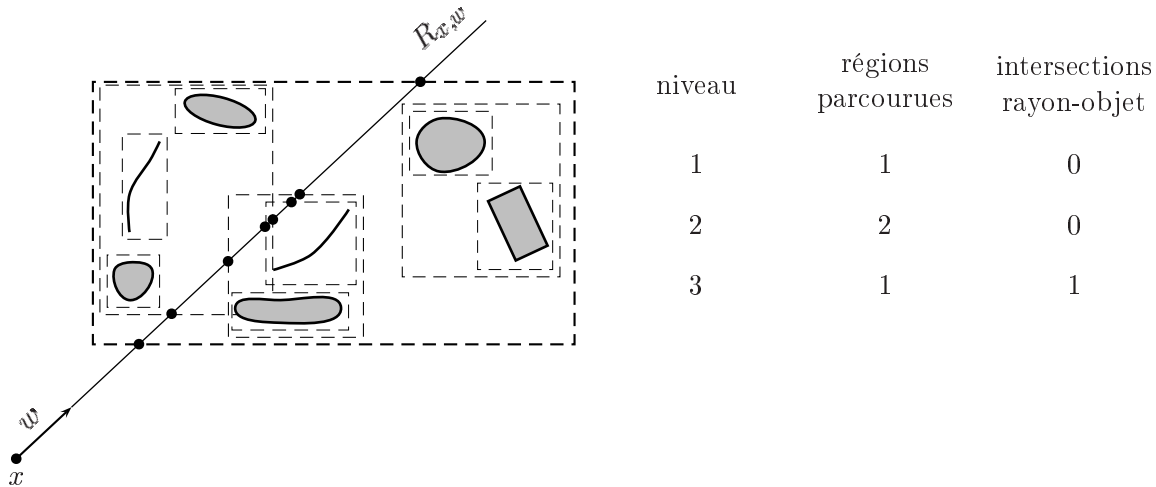


Fig. 3.3 – Hiérarchie d'objets englobants.

but de choisir l'objet englobant qui minimise cette complexité. Ce critère est utilisé pour chaque nœud de l'arbre.

Gervautz [42] s'intéresse au parcours du rayon. Il utilise la méthode de classification des points d'intersection introduite par Roth [69] en ne calculant des intersections que dans les régions situées dans un intervalle d'intersection défini à partir de la première intersection rencontrée avec un objet de la scène géométrique.

De manière similaire, Kay et Kajiya [55] utilisent la particularité de leurs objets englobants (des boîtes englobantes généralisées avec un nombre limité de directions) pour la recherche d'intersection. Les objets englobants sont classés dans un certain ordre défini par la distance entre l'origine du rayon considéré et le premier point d'intersection avec l'objet englobant considéré.

Charney et Scherson [16] améliorent l'algorithme de parcours du rayon présenté dans [55]. Le principe consiste à distinguer les régions qui contiennent l'origine du rayon considéré des autres régions. Cette distinction définit un ordre de parcours des nœuds.

Ke et Chang [56] vont plus loin dans l'amélioration de la méthode présentée dans [55]. Les objets englobants sont classés en fonction de leurs proximités relatives. Ainsi, les tests d'intersection entre les rayons et les objets englobants se font dans un ordre défini par cette classification, ce qui, statistiquement, réduit le nombre d'intersections à calculer. D'autre part, l'ensemble des objets englobants est projeté sur le plan image, ce qui permet d'éliminer un grand nombre de parties de l'image ne donnant pas lieu à des intersections (parties noires).

La principal inconvénient commun à toutes les variantes de ce type de méthode est que le parcours du rayon est dépendant de l'ordre dans lequel sont insérées dans la structure de données arborescente les diverses régions. C'est pourquoi l'efficacité de ces méthodes de partition est difficile à prédire. Goldsmith et Salmon [45] ont donné une formulation de la probabilité conditionnelle qu'un rayon heurte un objet sachant qu'il a heurté un objet l'englobant. Pour ce faire, ils ont supposé que les rayons sont distribués uniformément. La probabilité recherchée est alors égale au rapport des aires projetées moyennes (moyenne sur l'ensemble des directions des aires projetées le long d'une direction). Si de plus les objets considérés sont convexes (généralement les objets englobants sont convexes) alors la probabilité recherchée est le rapport des aires du contour des objets considérés :

$$P(R_{x,w} \cap O_{e,i} \neq \emptyset \mid R_{x,w} \cap O_e \neq \emptyset) = \frac{\text{aire}(\partial O_{e,i})}{\text{aire}(\partial O_e)}, \quad (3.5)$$

où $R_{x,w}$ est un rayon appartenant à l'ensemble des rayons considérés, $O_{e,i}$ est un objet convexe englobé par un objet convexe O_e et $\partial O_{e,i}$ et ∂O_e sont les contours respectifs des objets $O_{e,i}$ et O_e . Cette relation permet de donner une expression de la complexité en temps des méthodes utilisant des objets englobants

convexes. Soient $\{O_{e,i}\}_i$ des objets (objet de la scène géométrique ou objet englobant convexe) englobés par un objet convexe O_e . La complexité moyenne en temps de recherche de la première intersection entre un rayon et un objet contenu dans O_e s'écrit :

$$\bar{T}_{O_e} + \bar{T}_{\text{int},O_e} ,$$

où la complexité moyenne en temps intérieure \bar{T}_{int,O_e} s'exprime de manière récursive :

$$\bar{T}_{\text{int},O_e} = \sum_i \left(\bar{T}_{O_{e,i}} + \frac{\text{aire}(\partial O_{e,i})}{\text{aire}(\partial O_e)} \bar{T}_{\text{int},O_{e,i}} \right) , \quad (3.6)$$

où $\bar{T}_{O_{e,i}}$ est la complexité moyenne en temps de recherche de la première intersection avec l'objet $O_{e,i}$ et $\bar{T}_{\text{int},O_{e,i}}$ est la complexité moyenne en temps intérieure pour l'objet $O_{e,i}$. Par définition, la complexité moyenne en temps intérieure pour un objet de la scène géométrique est nulle. Cette estimation de la complexité moyenne en temps ne donne pas une mesure précise du coût dans un cas général. En particulier, la distribution de rayons est considérée uniforme, ce qui est faux dans la plupart des simulations. De plus, les phénomènes d'occlusions entre les objets englobés ne sont pas pris en compte. Cependant, l'utilisation de l'expression de la complexité moyenne en temps donne un moyen de construire une hiérarchie d'objets englobants qui consiste à minimiser l'expression 3.5 du rapport des aires des contours des objets englobants (pour des objets englobants convexes).

Le principe des méthodes spécifiques aux objets CSG est de définir, dans un premier temps, des boîtes englobantes minimales pour chaque objet de la scène géométrique. Une boîte englobante est généralement calculée en combinant les boîtes englobantes des objets primitifs à l'aide des opérateurs ensemblistes considérés pour l'objet (figure 3.4). La différence entre ce type de partition et une partition par hiérarchie d'objets englobants tient à la particularité des objets CSG. En effet, dans le cas présent, on ne cherche pas à localiser un objet mais plutôt une partie d'un objet, la plus petite possible. Cette partie est représentée par un sous-arbre CSG issu de l'objet considéré.

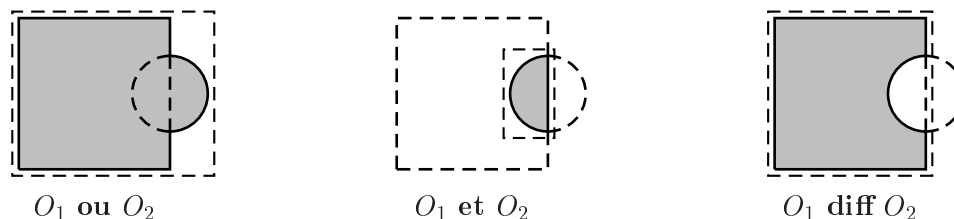


Fig. 3.4 – Boîte englobante minimale pour un objet CSG.

Arnaldi, Priol et Bouatouch [1] projettent les boîtes englobantes minimales de chaque objet sur le plan image. Une partition de type BSP (*Binary Space Partitionning*) est construite à partir des rectangles issus de la projection. À chaque segment d'un rectangle correspond un plan séparant l'espace de la scène géométrique en deux régions. Cette partition est raffinée à l'aide d'une subdivision le long de l'axe perpendiculaire au plan image. Les régions ainsi obtenues sont associées à une structure de données contenant notamment des informations de voisinage. Les rayons issus du plan image (dans le cas d'un lancer de rayons dans le sens inverse), c'est-à-dire les rayons primaires, utilisent principalement la première partition, tandis que les rayons secondaires se déplacent dans les régions issus de la seconde partition en utilisant les relations de voisinage entre les différentes régions. Pour éviter les calculs d'intersection redondants, ils ont mis en place sur chaque objet une information qui consiste à conserver le résultat du dernier calcul d'intersection avec un rayon. Ainsi, lorsque l'objet est rencontré plusieurs fois par un même rayon, le calcul n'est effectué qu'une seule fois (figure 3.5).

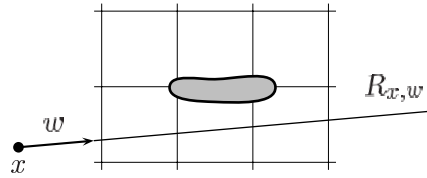


Fig. 3.5 – Réduction des calculs redondants lors du parcours du rayon.

Chuang et Hwang [17] utilisent des boîtes englobantes plus précises que [1], basées sur les boîtes englobantes de Cameron [13]. La partition s'appuie sur les faces des boîtes englobantes des objets. Une structure de données hiérarchique contient les liens entre les voxels et les sous-arbres CSG issus de la construction de la partition. Le parcours du rayon est de type itératif et utilise la classification des points introduite par Roth [69].

Méthodes globales

Ces méthodes consistent à partir de la scène géométrique. On considère un objet englobant la scène, généralement un parallélépipède. Pour être conforme aux notations du premier chapitre, notons X ce domaine. Une partition des objets de la scène est alors définie en subdivisant le domaine X en régions le plus souvent parallélépipédiques alors appelées voxels. La partition est alors un partitionnement de X en ce sens que les régions R_k sont disjointes et que leur union est l'ensemble X :

$$\begin{cases} \bigcup_{k=1}^n R_k = X \\ R_k \cap R_{k'} = \emptyset, k \neq k' \end{cases} .$$

À chaque région R_k est associée l'ensemble des objets qui ont une intersection avec cette région. Cet ensemble est caractérisé par la liste L_{R_k} donnée par la définition 6. Cette définition met en avant un des problèmes principal de la localisation, à savoir l'évaluation de l'intersection entre un objet et une région. Puisque l'on s'intéresse aux surfaces optiques (définition 5), pour les calculs d'intersection rayon-objet, l'idéal est donc de considérer, pour un objet O donné, la surface associée $\Sigma(O)$. Cependant, il n'est pas toujours facile d'évaluer si l'intersection entre une région et une surface est l'ensemble vide ou non. C'est pourquoi on a recours à un objet $\Upsilon(O)$ englobant O . Pour construire la liste associée à une région R_k , on utilise le critère d'exclusion géométrique suivant :

$$\Upsilon(O) \cap R_k = \emptyset \implies \Sigma(O) \cap R_k = \emptyset ,$$

qui résulte du fait que $\Sigma(O) \subset \Upsilon(O)$. La qualité de la localisation dépend alors des tailles des listes L_{R_k} . Celles-ci doivent naturellement être les plus petites possibles. La hiérarchisation de ces méthodes consistent à répéter la subdivision aux régions de la partition. Le critère d'arrêt le plus utilisé est un seuil définissant le nombre d'objets maximum pour une région et, bien entendu, une profondeur maximale de la hiérarchie.

Une fois la partition des objets définie, il reste à rechercher la première intersection entre un rayon et la scène géométrique considérée. Deux classes d'algorithmes sont associés à ce type de partition : les algorithmes itératifs et les algorithmes récursifs.

Différents types de partition ont été développées :

- binaire,
- octale,
- grille uniforme,
- macro-régions.

Le principe de la partition binaire, ou BSP (*Binary Space Partionning*), consiste à séparer par un plan chaque région considérée en deux sous-régions, en partant du domaine X (figure 3.6). Par nature, la structure de données associée à ce type de partition est l'arbre binaire.

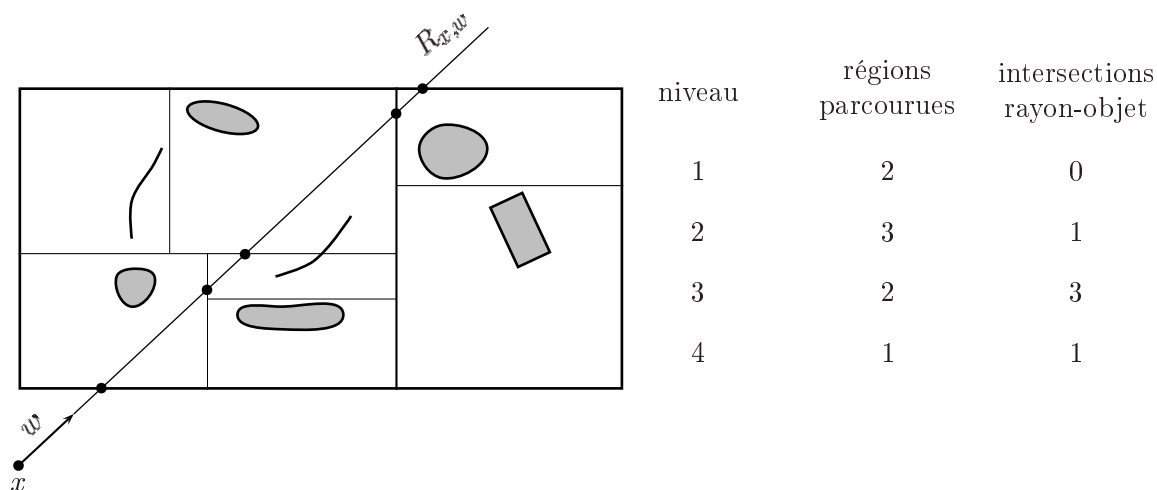


Fig. 3.6 – Partition binaire (cas de plans parallèles aux axes de la scène géométrique).

Kaplan [54] a introduit une méthode de partition binaire basée sur des plans parallèles aux axes de la scène géométrique. La structure de données est un arbre binaire classique et le parcours du rayon est de type itératif.

Jansen [49] a introduit une méthode similaire en ce qui concerne la partition et la structure de données associée, mais utilise un algorithme de parcours du rayon de type récursif dichotomique. L'arbre binaire associé à la partition est parcouru de haut en bas. Pour chaque nœud non terminal considéré, le parcours des deux fils est ordonné en fonction de l'ordre de parcours du rayon dans les deux régions associées.

Le principe de la partition octale, ou octree (qui signifie arbre octal), consiste à subdiviser le domaine X en huit régions disjointes, appelées octants. Chaque région est à nouveau subdivisée en huit octants. Et ainsi de suite jusqu'à un critère d'arrêt (figure 3.7). Par nature, la structure de données liée à ce type de partition est arborescente.

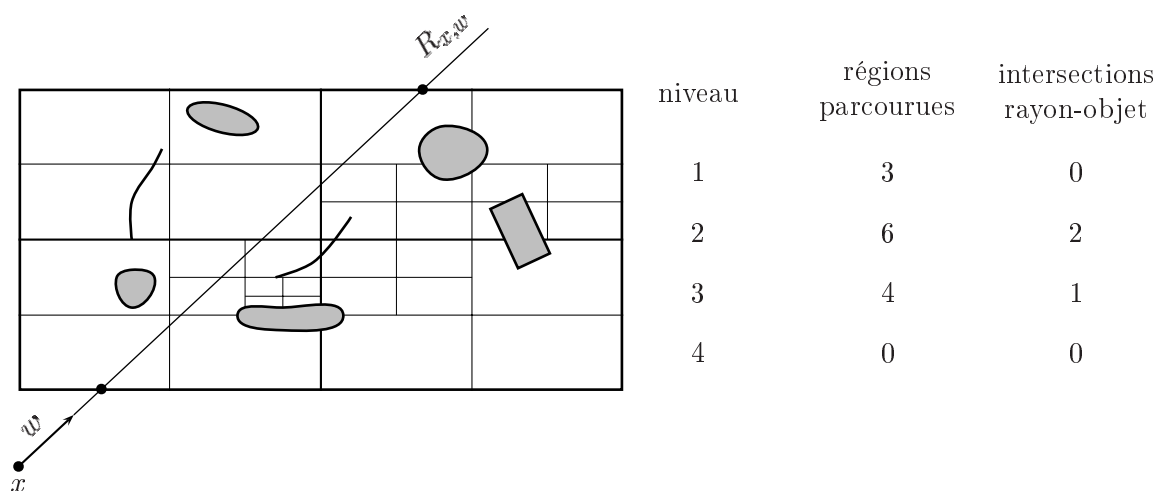
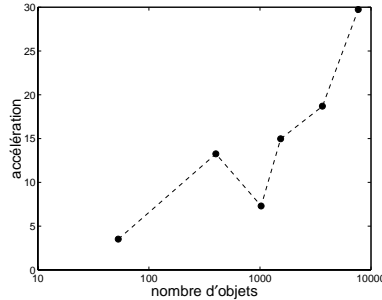


Fig. 3.7 – Partition octale (octants uniformes).

Glassner [43] fût un des premiers à mettre en place une telle partition, avec des octants uniformes.

La structure de données qu'il utilise est basée sur celle introduite par Meagher [66]. Le codage de l'octree est basé sur une numérotation suffixée des voxels. Le parcours du rayon consiste à calculer les intersections avec les faces des voxels voisins, puis à se déplacer le long du rayon d'une valeur suffisamment petite pour assurer que le point trouvé est bien dans le voxel suivant. La recherche du nœud dans l'arbre correspondant s'effectue depuis la racine, et utilise le codage particulier de l'octree. Le graphe ci-dessous présente le rapport des complexités en temps entre la méthode naïve et la méthode développée par Glassner pour des scènes contenant des nombres d'objets différents (d'après [43]).



Wyvill, Kunii et Shirai [88] ont utilisé une partition de type octree pour le modèle géométrique CSG. La construction de la partition est adaptative, les critères d'arrêts sont classiques (nombre d'objets primitifs dans un voxel et profondeur maximale de l'arbre associé à l'octree). L'information associée à chaque voxel est telle que la structure de données inclue l'arbre CSG de chaque objet. Ainsi, un nœud non terminal dans l'arbre de cette structure est associé soit à une feuille, soit à un sous-arbre CSG, soit à un octree. Une feuille est associée à un état parmi cinq : le voxel est vide, le voxel est contenu dans un objet primitif, le voxel est à cheval sur la contour d'un objet primitif (sans opérateur ensembliste **diff**), le voxel est à cheval sur la contour d'un objet primitif (avec opérateur ensembliste **diff**), le voxel est subdivisé. Le parcours du rayon est classique pour l'octree (Wyvill et Kunii [87]), un traitement particulier est appliqué aux nœuds associés à un sous-arbre CSG. Dans ce cas la recherche de la première intersection est obtenue par une méthode similaire à la méthode des epsilons présentée au paragraphe 2.2.4.

Whang, Song, Chang, Kim, Cho, Park et Song [85] définissent une partition de type octree, mais les octants ne sont pas nécessairement uniformes. Les auteurs utilisent le critère présenté dans [45] (expression 3.5) pour déterminer les octants qui rendent optimal la complexité en temps du lancer de rayon, au sens de l'expression 3.6.

Le principe de la partition en grille uniforme consiste à subdiviser chaque région considérée en un certain nombre de régions disjointes de même taille (figure 3.8). Les structures de données liées à ce type de subdivision sont des tableaux 3D, où chaque entrée correspond à un voxel, par le biais de trois coordonnées entières, et chaque case du tableau contient des informations, relatives à la géométrie et à la physique, liées au voxel considéré. Parmi ces informations, on trouve nécessairement les indices des objets.

Fujimoto, Tanaka et Iwata [41] furent les premiers à introduire ce type de partition avec une méthode qu'ils nomment ARTS (*Accelerating Ray Tracing System*). Le parcours d'un rayon est décrit à l'aide d'un algorithme connu sous le nom 3DDA (*3D Differential Digital Analyser*), qui est une généralisation de l'algorithme de Bresenham [11] (affichage d'une portion de droite sur un écran). Plus la subdivision est fine, plus le nombre d'objets à tester est petit. Cependant la complexité en temps du parcours du rayon et la complexité en mémoire du stockage de la structure de données augmentent avec la finesse de la subdivision.

Hsiung et Thibadeau [48] ont proposés une amélioration de la méthode, appelée FINE-ARTS (*Fast INDEXING hierarchically ENcoded ARTS*). Ils ont mis en place une structure de données alliant les avantages des structures liées aux octrees et des structures liées aux grilles uniformes. D'une part, la

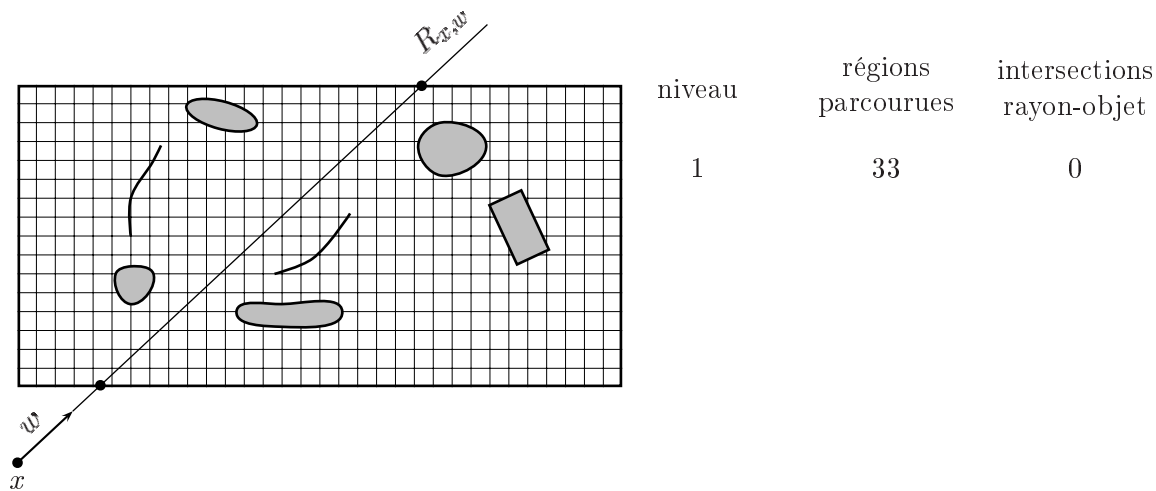


Fig. 3.8 – Grille uniforme.

complexité en temps de parcours du rayon est logarithmiquement proportionnelle à la finesse de la subdivision, alors que dans ARTS elle est linéairement proportionnelle. D'autre part, la complexité en mémoire pour le stockage des liens entre les voxels et les objets croît de manière linéaire par rapport à la finesse de la subdivision.

Cohen et Sheffer [19] proposent une méthode pour accélérer le parcours d'un rayon dans une partition en grilles uniformes (figure 3.9). Elle consiste à parcourir le plus vite possible les espaces vides. Pour ce faire, une mesure de la proximité entre les voxels vides et les voxels contenant des objets est stockée en mémoire. Un rayon se trouvant dans un voxel vide n'a pas d'intersection avec un objet dans un certain intervalle dont la longueur est déterminée par la mesure de proximité propre au voxel considéré. On peut ainsi se déplacer le long du rayon de la distance concernée.

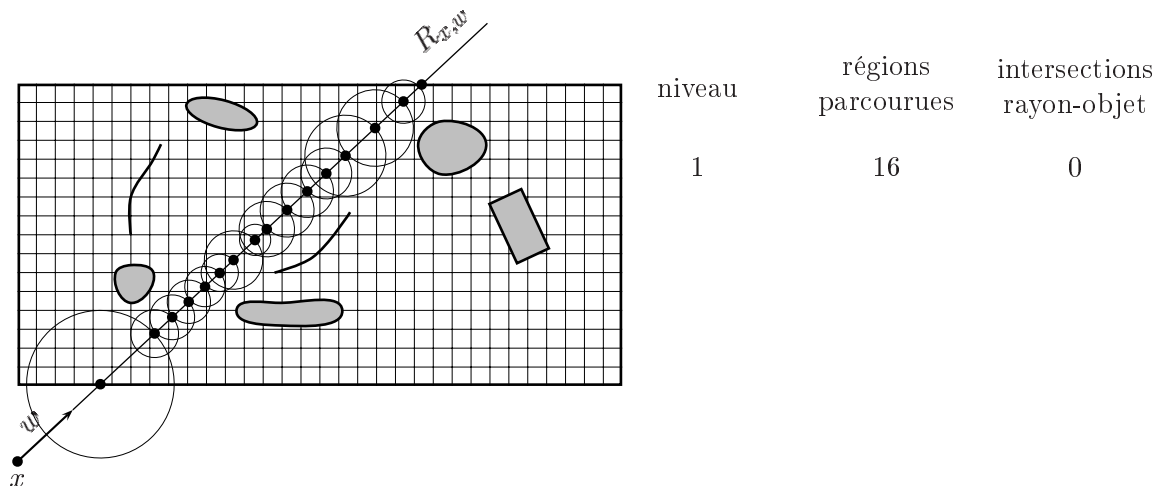


Fig. 3.9 – Méthode de proximité (distance euclidienne).

Yagel, Cohen et Kaufman [89] présentent une approche originale, appelée RRT (*Raster Ray Tracer*), basée sur la partition en grille uniforme, et dont l'originalité réside dans la nature de l'information associée à chaque voxel, qui se traduit par une discrétisation de la scène géométrique (figure 3.10). La méthode se décompose en deux phases. Tout d'abord, la scène est approchée par un ensemble de voxels. Chaque voxel contient de l'information qui représente localement les propriétés géométrique (vecteur normal) et physiques (couleur, opacité, texture) de chaque objet. Le parcours de rayons est basé sur la technique classique de type 3DDDA, c'est un rayon discret (6 ou 26 voisins selon l'algorithme choisi)

qui est propagé. Le rayon s'arrête dès qu'il rencontre un voxel non vide et l'interaction avec le rayon est calculée en fonction des données associées au voxel rencontré. Le principal intérêt de la méthode est qu'il n'y a pas de calcul d'intersection rayon-objet. Les calculs sont d'autant plus rapides qu'ils sont effectués en arithmétique entière.

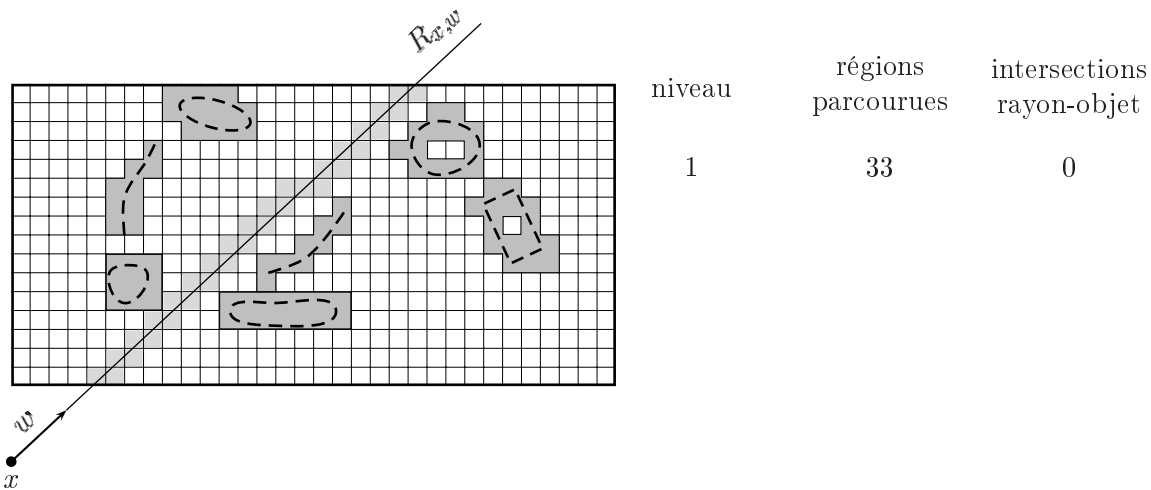


Fig. 3.10 – Lancer de rayons discret.

Cette méthode pose de nombreux problèmes. D'une part, la discrétisation des objets est un problème difficile, d'autant plus qu'un voxel est lié à un seul objet, ce qui pose des problèmes de cohérence géométrique, par exemple dans le cas où plusieurs objets se trouvent dans le même voxel. D'autre part, cette discrétisation implique des calculs d'intersection rayon-objet incorrects. De plus, pour obtenir une qualité du résultat acceptable (par rapport au problème d'aliassage principalement), il faut construire une subdivision la plus fine possible ce qui requiert une grande quantité de mémoire et implique un coût de parcours du rayon élevé. Delfosse, Hewitt et Mériaux [30] proposent une méthode, appelée SRDRT (*Restricted Semi Discrete Ray Tracing*), qui est basée sur la méthode RRT, avec des calculs réels d'intersection rayon-objet et de vecteur normal, dans le cas particulier de scènes polygonales, en ajoutant des informations topologiques visant à résoudre les problèmes de cohérence géométrique. Cette méthode se rapproche de la méthode ARTS. Ainsi la qualité des images obtenues et l'efficacité de l'algorithme sont intéressantes.

Deville [34] propose une méthode basée sur la partition en grille uniforme dont l'objectif est de diminuer le coût du parcours du rayon. La méthode des macro-régions consiste à partir d'une grille uniforme, puis à regrouper les voxels voisins ne contenant aucun objet (figure 3.11). Un regroupement s'appelle une macro-région, qui n'est autre qu'un parallélépipède. Chaque voxel vide est associé à une macro-région (la plus grande). Lors du parcours du rayon, lorsqu'un voxel vide lié à une macro-région est rencontré, alors celle-ci est directement traversée. Cette méthode a donc pour but réduire le nombre de voxels à parcourir.

Méthodes mixtes

Ces méthodes utilisent les avantages des deux classes de méthodes précédentes en essayant de s'affranchir au mieux des inconvénients. En effet, les méthodes locales ont un pouvoir séparant de très bonne qualité puisque chaque région considérée, au début de la mise en place de la construction de la partition, contient exactement un objet. D'un point de vue géométrique, ce n'est pas vrai puisqu'il peut y avoir des phénomènes de recouvrement entre les différentes régions, mais, d'un point de vue de la structure de données, c'est-à-dire au regard de la partition de l'ensemble des objets, c'est bien vrai. Le point faible de ces méthodes est la parcours du rayon. En effet, il est fortement dépendant de

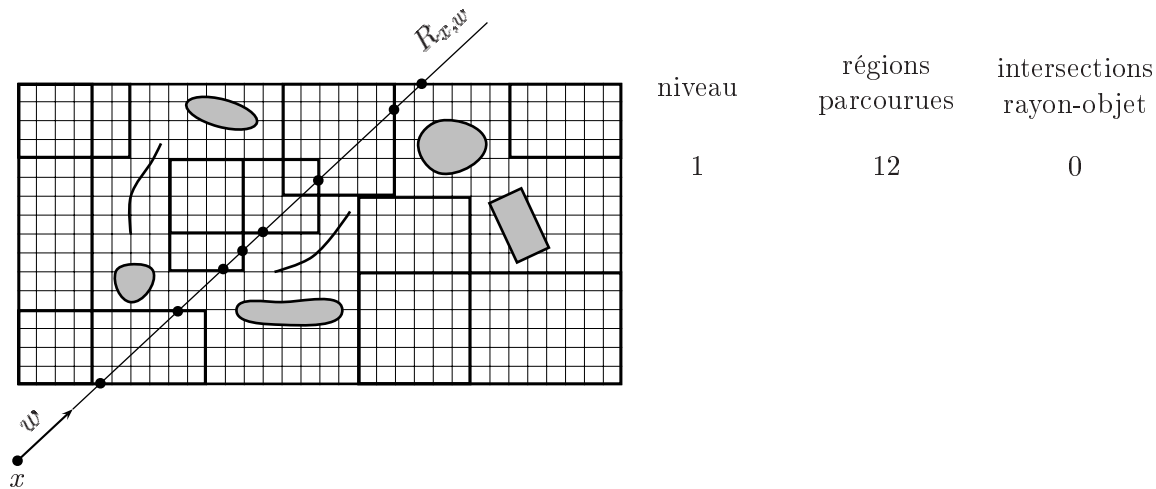


Fig. 3.11 – Macro-régions.

la manière dont les régions sont regroupées. Les méthodes globales offrent un parcours du rayon très simple et dont le coût est relativement faible. Cependant, ces méthodes souffrent d'une grande difficulté à bien séparer les objets, sans trop raffiner la partition, et ont des difficultés à parcourir rapidement les espaces vides. À la lueur de ces remarques, le principe des méthodes mixtes consiste, dans un premier temps, à utiliser une méthode locale. Les objets sont regroupés dans des régions selon principalement deux critères : la dimension relative des objets et leur distance relative. On obtient ainsi une première partition de la scène géométrique. Dans un second temps, sur chaque région de cette partition est alors appliquée une méthode de partition globale sur chaque sous-scène géométrique contenue dans la région considérée.

Le parcours du rayon se décompose alors en deux parties. Un parcours des régions issues de la première partition et un parcours des sous-régions issues des secondes partitions.

Snyder et Barr [80] furent les premiers à mettre en place une partition à l'aide d'une hiérarchie d'objets englobants et de grilles uniformes (figure 3.12). L'objectif des auteurs était de traiter des scènes géométriques contenant des surfaces triangulées contenant jusqu'à $4 \cdot 10^{11}$ triangles. Pendant la construction de la partition, les objets de la scène géométrique et les régions déjà définies sont soit insérés dans des boîtes englobantes, sur lesquelles sont utilisées des grilles uniformes, soit regroupées dans des boîtes englobantes. Généralement, au départ de la construction, chaque surface est insérée dans une boîte englobante subdivisée à l'aide d'une grille uniforme.

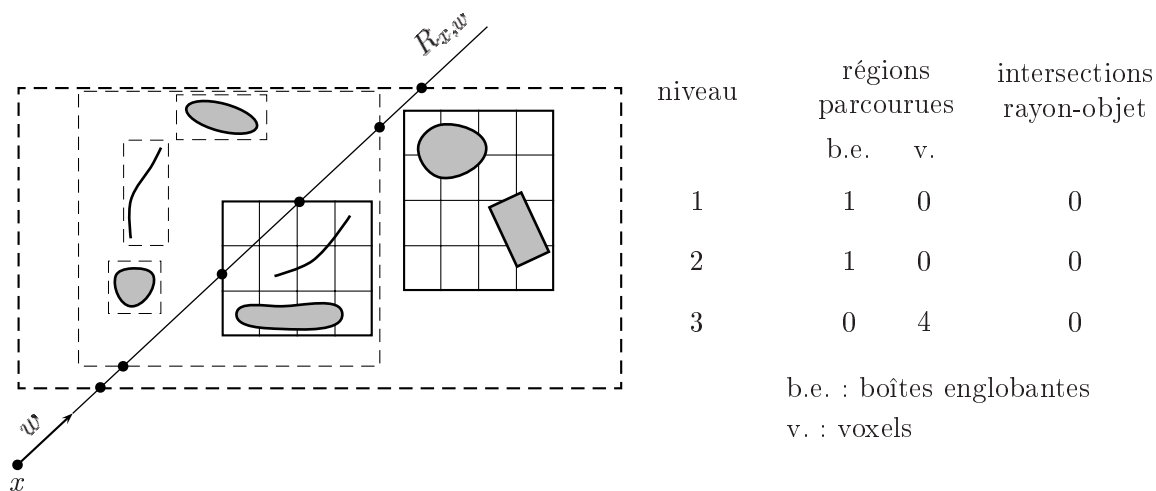


Fig. 3.12 – Hiérarchie de boîtes englobantes et de grilles uniformes.

Pour le parcours, les auteurs proposent deux améliorations des méthodes classiques : une pour le parcours des grilles uniformes et une pour le parcours de boîtes englobantes. Dans les deux cas, l'objectif est d'accélérer le parcours des espaces vides (figure 3.13). Pour les boîtes englobantes le rayon définit une boîte d'exclusion. Pour les grilles uniformes, les boîtes englobantes sont utilisées comme critère d'exclusion dans les sous-voxels.

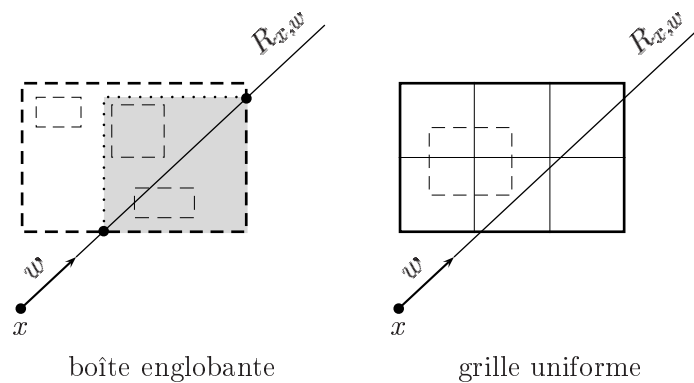


Fig. 3.13 – Accélération du parcours du rayon dans une boîte englobante et dans une grille uniforme.

Cazals, Drettakis et Puech [15] s'intéressent principalement aux scènes contenant une grande quantité d'objets. Les auteurs proposent une partition, appelée HUG (*Hierarchy of Uniform Grids*), dont un des intérêts principal est l'automatisation de la construction de la structure de données associée. Pour ce faire, une première partition est mise en œuvre, pour laquelle les objets sont tout d'abord filtrés selon un critère de taille, puis regroupés selon un critère de distance dans des régions parallélépipédiques appelées clusters. Sur chacun de ces clusters est alors appliquée une seconde partition de type grille uniforme. Une hiérarchie de ces clusters, appelée hiérarchie de grilles uniformes, est alors mise en place (figure 3.14). Chaque cluster est alors soit considéré comme un objet, soit comme un sous-cluster d'un autre cluster. L'algorithme de parcours du rayon comprend une partie itérative, pour le parcours des grilles uniformes, englobée dans une partie réursive, pour le passage à une grille située à un autre niveau de la hiérarchie.

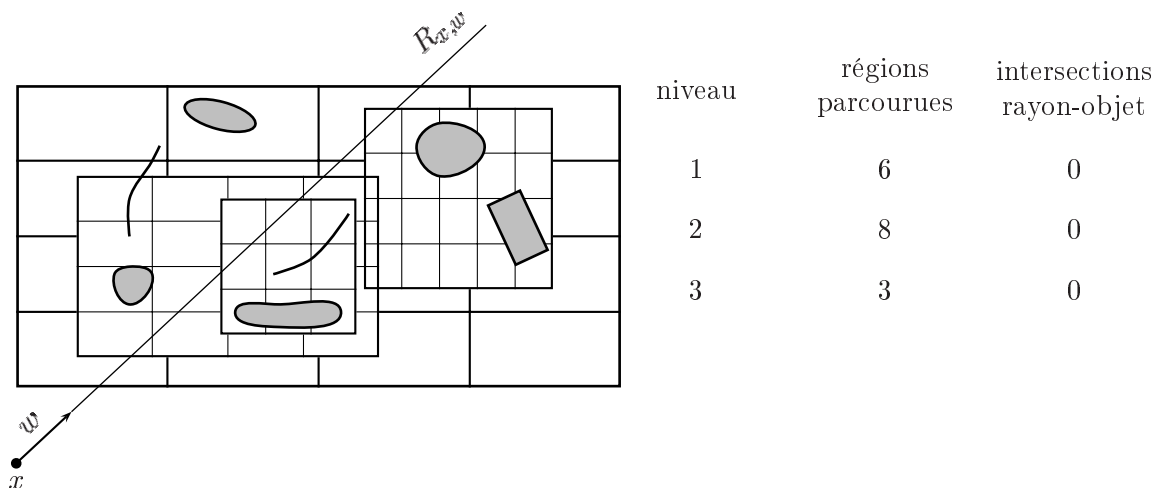


Fig. 3.14 – Hiérarchie de grilles uniformes.

Klimaszewski et Sederberg [59] présentent une méthode hybride similaire à HUG, basée sur la méthode de partition présentée dans [80]. Les auteurs vont plus loin en recherchant des critères de construction

de la partition, dans le but d'adapter au mieux les grilles uniformes aux caractéristiques locales de la scène géométrique. Dans un premier temps, une hiérarchie de boîtes englobantes est mise en place en utilisant l'algorithme introduit par Goldsmith et Salmon [45]. Cette construction est effectuée à partir d'un critère consistant à minimiser l'aire du contour des boîtes englobantes (expression 3.5). Dans un second temps, sur chaque boîte englobante est mis en place une partition à l'aide d'une grille uniforme, construite à partir d'un critère qui prend en compte le nombre d'objets (les objets de la scène géométrique et les boîtes englobantes) et les dimensions de la boîte englobante considérée. Enfin, une hiérarchie des grilles uniformes est mise en place, à l'aide d'une amélioration de la méthode présentée par Jevans et Wyvill [50]. Les sous-grilles s'appuient sur des boîtes minimales et non sur les voxels de la grille à subdiviser, ce qui permet de réduire le coût de traversée des espaces vides par des rayons (figure 3.15). L'algorithme de parcours du rayon est adapté en conséquence.

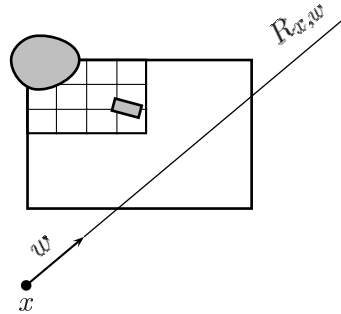


Fig. 3.15 – *Sous-grille réduite et parcours du rayon.*

3.1.4 Choix d'une méthode

Pour choisir un type de méthode d'accélération des calculs il faut commencer par définir le besoin. Ce travail doit s'effectuer dans un contexte industriel et l'intégration de la méthode dans un environnement logiciel existant. Le cahier des charges comporte les points suivants.

- Le temps de développement est limité.
- Les structures de données et les algorithmes développés dans le cadre de ce travail doivent être propre au contexte de développement (pas de récupération de code existant).
- La puissance des machines est limitée par le type de matériel utilisé par les utilisateurs des logiciels (machines mono-processeur, pas de réseau, système d'exploitation des machines fixé, etc.).
- Le modèle géométrique utilisé dans les logiciels concernés est hybride. Une partie est de type CSG, l'autre de type BRep.
- Les systèmes optiques considérés sont quelconques (petits ou grands objets, répartition des objets quelconque, nombre d'objets quelconque, etc.).
- La méthode d'accélération à implémenter doit respecter la géométrie et la physique (aucune approximation n'est envisageable).
- L'utilisateur du logiciel ne doit pas intervenir manuellement pendant le calcul.

L'objectif du travail est bien entendu l'amélioration des performances des logiciels concernés, qui se dégradent très vite dès que l'on souhaite augmenter la précision des résultats. La méthode doit se mettre en place dans un contexte très précis. Elle doit donner lieu à un module devant s'insérer dans l'architecture de développement existante (figure 3.16).

Le module de calcul utilise le lancer de rayons, dont les algorithmes principaux sont présentés dans les sections 2.2 et 2.3. Dans le modèle CSG une accélération est proposée pour le calcul d'intersection rayon-objet. C'est une méthode utilisant des sphères englobantes, sans hiérarchisation. Dans le modèle BRep

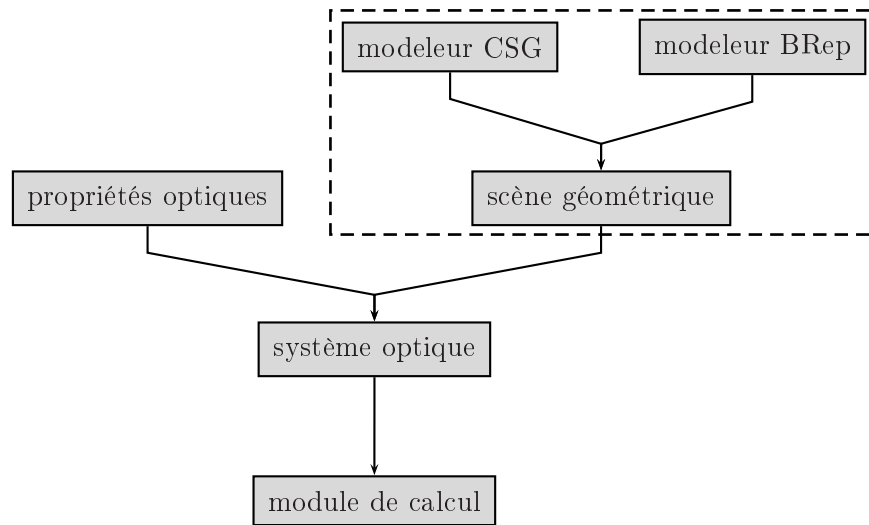


Fig. 3.16 – Contexte de développement.

les calculs d'intersection ne sont pas accessibles. L'amélioration des performances, en tenant compte du cahier des charges, ne peut donc pas passer par des méthodes d'intersection rayon-objet spécifiques. Le choix s'oriente donc vers une méthode de localisation spatiale par hiérarchie de subdivisions uniformes. Plusieurs raisons à cela, en réponse au cahier des charges essentiellement.

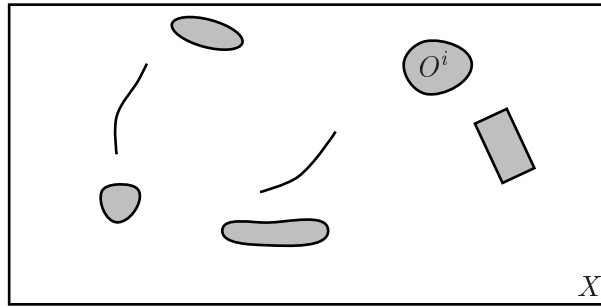
- La mise en œuvre d'une telle méthode se fera dans un temps de développement raisonnable.
- Les structures de données développées seront spécifiques, et l'encombrement mémoire sera limitée, ce qui permettra une utilisation sur le matériel envisagé.
- L'adaptation de la méthode pour les modèles CSG et BRep ne posera pas de problème particulier, du moment que l'on possède certaines fonctions géométriques particulières à chaque modèle (intersection d'une surface avec une face plane notamment).
- La géométrie sera respectée, aucune approximation n'est nécessaire.
- Une méthode adaptative permettra de prendre en compte la diversité des systèmes optiques envisageables, et ne nécessitera aucune intervention manuelle pendant le calcul.
- Les méthodes basées sur une partition de l'espace en subdivisions uniformes sont parmi les plus performantes.

3.2 Localisation par subdivision spatiale

L'environnement dans lequel doit être intégré le module d'accélération des calculs comporte un modèleur géométrique. Un système optique est donc défini par des objets de l'un ou l'autre des modèles. Pour être conforme avec les notations du deuxième chapitre, on notera O^1, \dots, O^N l'ensemble des objets du système optique considéré.

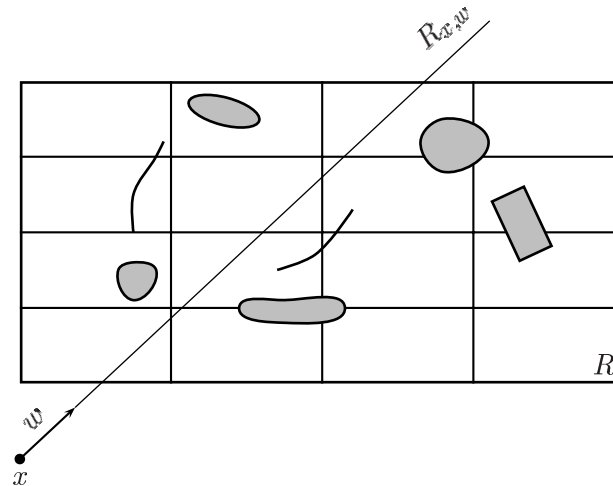
Le modèle physique, présenté dans le premier chapitre, est défini dans un certain domaine fermé X . Dans ce paragraphe, et dans la suite du chapitre, le domaine X considéré est un parallélépipède parallèle aux axes, à partir duquel la localisation spatiale va être considérée (figure 3.17).

Dans la suite nous présentons la méthode développée dans le contexte industriel. Pour ce faire, on procède en deux étapes. Le premier paragraphe décrit les techniques nécessaires à la mise en œuvre d'une subdivision uniforme. Le second décrit la technique de hiérarchisation des subdivisions uniformes.

Fig. 3.17 – *Domaine englobant la scène géométrique.*

3.2.1 Subdivision uniforme

Une subdivision uniforme d'une région parallélépipédique R , appelée voxel, est une partition en régions de dimensions identiques (figure 3.18). C'est ce qu'on appelle aussi une grille uniforme. Soit $\mathcal{S} = \{R_k\}_{k=1}^n$ l'ensemble des régions d'une subdivision uniforme donnée.

Fig. 3.18 – *Subdivision uniforme.*

La mise en place d'une telle méthode comporte deux phases :

- une phase de pré-calcul consistant à localiser les objets dans la subdivision \mathcal{S} ,
- une phase de calcul consistant à définir le parcours d'un rayon dans la subdivision \mathcal{S} .

Localisation des objets

Les hypothèses faites sur le modèle physique font que l'on s'intéresse uniquement aux intersections entre des rayons et des surfaces optiques². Ce sont les $\Sigma(O^i)$, c'est-à-dire, dans le contexte qui nous intéresse, les contours des objets CSG et les faces des objets BRep. Il est nécessaire de créer une structure de données qui lie les régions $\{R_k\}_k$ aux objets $\{O^i\}_i$, c'est-à-dire se donner un moyen de construire les listes L_{R_k} données par la définition 6 :

$$L_{R_k} = \{i \in \{1, \dots, N\}, \Upsilon(O^i) \cap R_k \neq \emptyset\} ,$$

où l'on rappelle que $\Upsilon(O^i)$ représente l'objet considéré pour l'évaluation de l'intersection avec R_k . La définition de ces listes permet de différencier les types de localisation d'un objet dans une subdivision

²On rappelle que les surfaces optiques (définition 5 page 96) sont le lieu de discontinuité de l'indice de réfraction optique (chapitre 1 page 15).

uniforme. Soit la localisation est effectuée sur la surface optique ($\Upsilon(O^i) = \Sigma(O^i)$). Soit la localisation est effectuée sur un objet englobant, le plus simple étant une boîte englobante ($\Upsilon(O^i) = B(O^i)$). En pratique le choix du type de localisation dépend de la nature des objets considérés. Pour certains objets, il existe des méthodes qui permettent d'évaluer l'intersection entre une surface optique et une région (figure 3.19). L'utilisation des boîtes englobantes des objets fournit un critère d'exclusion simple et très rapide à évaluer. Ainsi, pour un objet O^i donné, ne seront testées que les régions qui ont une intersection avec la boîte englobante de l'objet, notée $B(O^i)$:

$$B(O^i) \cap R_k = \emptyset \implies \Upsilon(O^i) \cap R_k = \emptyset .$$

Cependant, l'évaluation de $\Sigma(O^i) \cap R_k$ est en général un problème difficile, et dans bien des cas se limite à un test avec les boîtes englobantes. Il ne faut pas perdre de vue que l'on ne doit pas se tromper lors de cette évaluation, en ce sens qu'il ne faut pas exclure une surface optique d'une région avec laquelle il y a bien intersection. C'est ce qu'on appelle le critère de robustesse.

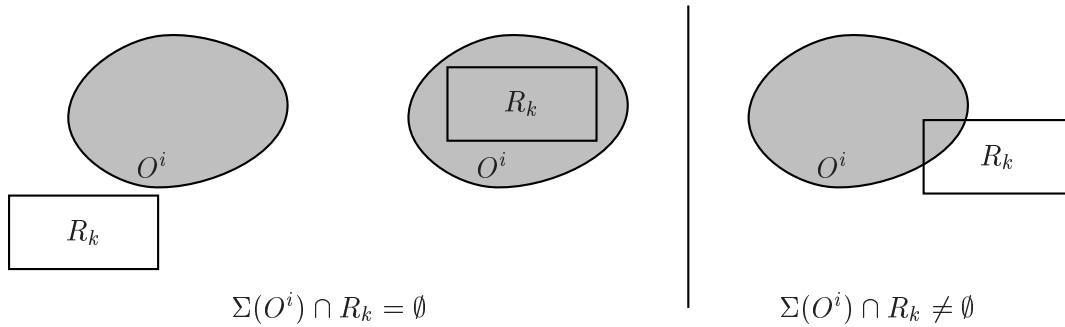


Fig. 3.19 – *Intersection entre une surface optique (cas d'un objet volumique sur le dessin) et une région.*

Pour les objets CSG implicites, il est intéressant en premier lieu de leur associer des boîtes englobantes minimales. Ensuite, la nature implicite des objets primitifs nous incite à considérer deux méthodes pour la localisation. Soit on approche la surface optique par un ensemble de voxels à l'aide d'une subdivision non uniforme (Khachan [57]), et on teste l'intersection entre les régions issues de R et les voxels issus de la surface optique. Soit on recherche les solutions du système semi-algébrique associé à l'objet CSG considéré dans la région considérée (Marchepoil [64]). Pour ce faire, on suppose que les polynômes des systèmes semi-algébriques sont exprimés dans la base de Bernstein (si besoin on peut effectuer un changement de base), dont les propriétés font que le critère de robustesse est satisfait. Cependant, on doit garder à l'esprit que le changement de base polynomiale n'est pas sans poser de problèmes numériques (Daniel [24]).

Pour les objets BRep paramétriques, là aussi il est intéressant de leur associer des boîtes englobantes minimales. Ensuite, si l'on souhaite une localisation plus précise, basée sur les surfaces, alors on ne sait pas faire de manière générale sous la contrainte du critère de robustesse. En effet, les méthodes numériques classiques sont de type suivi de contour. Par conséquent, on adaptera les méthodes aux types d'objets considérés. Par exemple, on pourra approcher les surfaces à faible courbure par des facettes planes et localiser ces facettes avec une certaine tolérance.

Une structure de données doit être mise en place pour le stockage en mémoire des liens entre les objets et les régions (figure 3.20). Pour répondre au cahier des charges elle doit satisfaire principalement deux exigences. D'une part, son encombrement en mémoire doit être le plus petit possible, car la capacité des machines est limitée. C'est pourquoi les régions vides (celles qui n'ont d'intersection avec aucun objet) ne seront pas enregistrées dans la structure de données, ce qui réduit l'information des liens aux listes suivantes :

$$\{L_{R_k}, L_{R_k} \neq \emptyset\} .$$

D'autre part, un ordonnancement particulier de l'information dans la structure de données permet d'utiliser des algorithmes de recherche de type dichotomique, de manière à ne pas dégrader les performances.

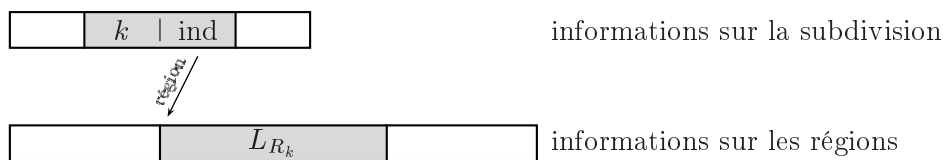


Fig. 3.20 – Structure de données pour l'enregistrement des liens entre les objets et les régions.

Parcours du rayon

La description du parcours d'un rayon dans une subdivision uniforme nécessite dans un premier temps de repérer les régions dans la subdivision. Le repérage s'effectue par un indice ou par des coordonnées discrètes. Par convention, à la région R_k sont associées les coordonnées discrètes $(c_0(k), c_1(k), c_2(k))$ (figure 3.21). La coordonnée dans la direction i est dans l'intervalle $[0, n_i - 1]$, où n_i est le nombre de subdivisions dans cette direction.

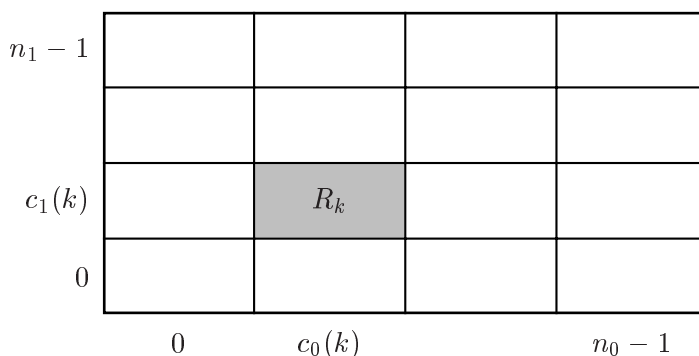


Fig. 3.21 – Système de coordonnées dans une subdivision uniforme.

Le passage des coordonnées $(c_0(k), c_1(k), c_2(k))$ à l'indice k est trivial :

$$k = c_0(k) + n_0 c_1(k) + n_0 n_1 c_2(k) .$$

Le passage inverse s'écrit :

$$\begin{cases} c_2(k) = k \operatorname{div} (n_0 n_1) \\ c_1(k) = (k \operatorname{mod} (n_0 n_1)) \operatorname{div} n_0 \\ c_0(k) = (k \operatorname{mod} (n_0 n_1)) \operatorname{mod} n_0 \end{cases} ,$$

où div est la division entière et mod est le reste de la division entière.

Le parcours d'un rayon dans une subdivision uniforme est effectué à l'aide d'un algorithme dont l'origine est l'affichage d'une portion de droite sur un écran (Bresenham [10]). Cette méthode a été généralisée par Slater [79] qui considère des régions non nécessairement cubiques et un espace de propagation de dimension quelconque. Il se décompose en deux étapes :

- recherche de la première région traversée,
- parcours des régions.

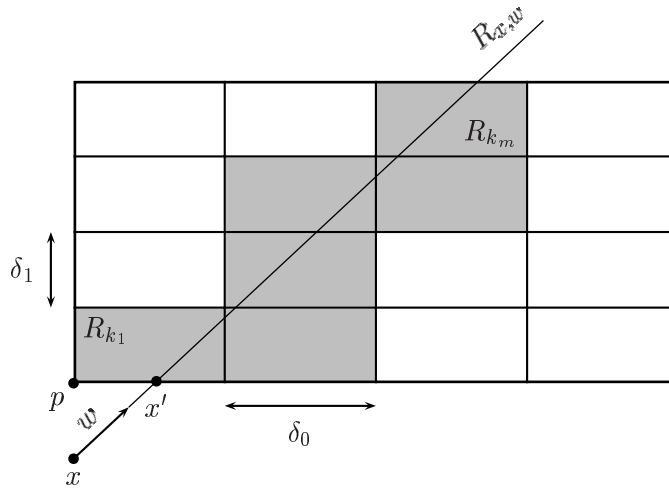


Fig. 3.22 – Parcours du rayon dans une subdivision uniforme.

Soit $R_{x,w}$ un rayon. Notons R_{k_1}, \dots, R_{k_m} la liste des régions successivement rencontrées par le rayon (figure 3.22).

Les coordonnées de la première région rencontrée par le rayon, c'est à dire R_{k_1} , sont données par :

$$c_i(k_1) = \left\lfloor \frac{x'_i - p_i}{\delta_i} \right\rfloor, \forall i \in \{0, 1, 2\},$$

où x' est le premier point d'intersection entre le rayon et le voxel R , p est le point inférieur du voxel R , et les δ_i sont les longueurs dans chaque direction des régions.

L'algorithme PREMIEREREGION recherche, si elle existe, la première région rencontrée par le rayon et retourne les coordonnées de celle-ci ainsi que le premier point d'intersection entre le rayon et le voxel R (algorithme 10).

Algorithme 10: Recherche de la première région rencontrée par un rayon.

PREMIEREREGION(\mathcal{S}, x, w)

- (1) $t \leftarrow \text{INTERRAYONVOXEL}(x, w, R)$
- (2) **if** $t < +\infty$
- (3) $fin \leftarrow \text{false}$
- (4) $x' \leftarrow x + tw$
- (5) **for** $i = 0$ **to** 2
- (6) $c_i \leftarrow \lfloor (x'_i - p_i) / \delta_i \rfloor$
- (7) **else**
- (8) $fin \leftarrow \text{true}$
- (9) **return** ($fin, (c_0, c_1, c_2), x'$)

L'algorithme INTERRAYONVOXEL retourne le paramètre sur le rayon de la première intersection entre un rayon et un voxel. Par convention, s'il n'y a pas d'intersection alors la valeur retournée est $+\infty$.

Désormais, supposons connue la suite des voxels traversés par le rayon jusqu'à un certain voxel R_{k_j} ($1 \leq j < m$) et cherchons le voxel suivant $R_{k_{j+1}}$ (figure 3.23).

Pour chaque direction $i \in \{0, 1, 2\}$, on recherche l'intersection entre le rayon $R_{x,w}$ et le plan d'équation $y_i = p_i + (c_i(k_j) + \text{pos}(w_i)) \delta_i$, c'est-à-dire le point y défini par :

$$\begin{cases} y_i = p_i + (c_i(k_j) + \text{pos}(w_i)) \delta_i \\ y_i = x'_i + t_i w_i \end{cases},$$

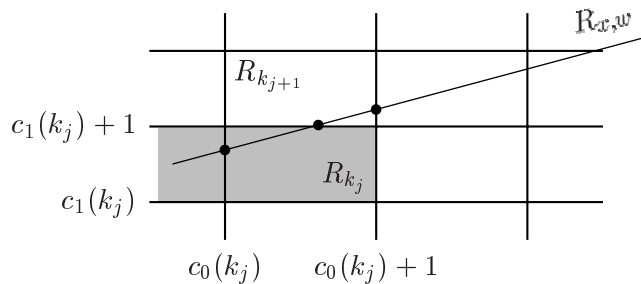


Fig. 3.23 – Recherche de la région suivante.

où $\text{pos}(w_i)$ est, par convention sur le repérage des régions, définie de la manière suivante :

$$\text{pos}(w_i) = \begin{cases} 1 & \text{si } w_i > 0 \\ 0 & \text{sinon} \end{cases} .$$

Les $\{t_i\}_i$ sont par conséquent donnés par :

$$t_i = \begin{cases} \frac{p_i + (c_i(k_j) + \text{pos}(w_i)) \delta_i - x_i}{w_i} & \text{si } w_i \neq 0 \\ +\infty & \text{sinon} \end{cases} .$$

Soit alors $i_{\min} \in \{0, 1, 2\}$ la direction correspondant au plus petit paramètre le long du rayon :

$$t_{i_{\min}} = \min_i \{t_i\} .$$

Il se peut que plusieurs directions satisfassent l'expression précédente, c'est le cas d'un rayon qui passe par un sommet de la région R_{k_j} . Dans cette situation, on peut considérer une de ces directions ou avancer dans toutes les directions concernées en même temps (figure 3.24). En géométrie discrète, les deux considérations précédentes correspondent respectivement aux voisinages 26-connexes et 6-connexes (en dimension 3).

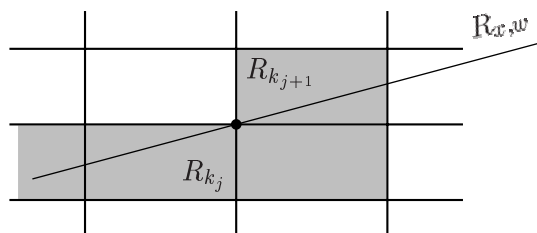


Fig. 3.24 – Cas où plusieurs directions sont candidates pour la recherche de la région suivante.

Les coordonnées de la région $R_{k_{j+1}}$ sont alors données par :

$$c_i(k_{j+1}) = \begin{cases} c_i(k_j) & \text{si } i \neq i_{\min} \\ c_i(k_j) + \text{sgn}(w_i) & \text{sinon} \end{cases} ,$$

où $\text{sgn}(w_i)$ est le signe de w_i :

$$\text{sgn}(w_i) = \begin{cases} -1 & \text{si } w_i < 0 \\ 0 & \text{si } w_i = 0 \\ 1 & \text{si } w_i > 0 \end{cases} .$$

Il reste à définir un critère d'arrêt du parcours, qui correspond à la sortie du rayon de la subdivision. Lorsque toutes les régions sont connues jusqu'à la région R_{k_m} , alors la région $R_{k_{m+1}}$ vérifie nécessairement une des deux conditions suivantes :

$$\begin{cases} c_{i_{\min}}(k_{m+1}) < 0 & \text{si } w_{i_{\min}} < 0 \\ c_{i_{\min}}(k_{m+1}) > n_{i_{\min}} & \text{si } w_{i_{\min}} > 0 \end{cases} ,$$

où $n_{i_{\min}}$ est le nombre de subdivisions dans la direction i_{\min} .

L'algorithme REGIONSUIVANTE recherche, s'il existe, la région suivante traversée par le rayon (algorithme 11).

Algorithme 11: *Passage à la région suivante.*

REGIONSUIVANTE($\mathcal{S}, (c_0, c_1, c_2), x', w$)

```
(1)   $t_{\min} \leftarrow +\infty$ 
(2)  for  $i = 0$  to 2
(3)    if  $w_i \neq 0$  then  $t \leftarrow (p_i + (c_i + \text{pos}(w_i)) \delta_i - x'_i) / w_i$ 
(4)    else  $t \leftarrow +\infty$ 
(5)    if  $t < t_{\min}$ 
(6)       $t_{\min} \leftarrow t$ 
(7)       $i_{\min} \leftarrow i$ 
(8)   $fn \leftarrow \text{false}$ 
(9)   $i \leftarrow 0$ 
(10) while not  $fn$  and  $i \leq 2$ 
(11)   if  $i = i_{\min}$ 
(12)      $c_i \leftarrow c_i + \text{sgn}(w_i)$ 
(13)      $fn \leftarrow (c_i < 0)$  or  $(c_i > n_i)$ 
(14)      $i \leftarrow i + 1$ 
(15) return  $(fn, (c_0, c_1, c_2))$ 
```

L'algorithme PARCOURSSUBUNIFORME consiste alors à rechercher, s'il existe, la première région rencontrée, à l'aide de l'algorithme PREMIEREREGION, puis à parcourir les régions à l'aide de l'algorithme REGIONSUIVANTE jusqu'à trouver une intersection avec un objet ou sortir de la subdivision (algorithme 12).

Algorithme 12: *Parcours d'un rayon dans une subdivision uniforme.*

```

PARCOURSUBUNIFORME( $\mathcal{S}, x, w$ )
(1)  ( $i_{\text{obj}}, t_{\text{obj}}, I_{\text{obj}}$ )  $\leftarrow (-1, +\infty, \{\})$ 
(2)  ( $fin, (c_0, c_1, c_2), x'$ )  $\leftarrow$  PREMIEREREGION( $\mathcal{S}, x, w$ )
(3)  while not  $fin$ 
(4)    ( $i_{\text{obj}}, t_{\text{obj}}, I_{\text{obj}}$ )  $\leftarrow$  TRAITERREGION( $\mathcal{S}, (c_0, c_1, c_2), x, w$ )
(5)    if  $t_{\text{obj}} < +\infty$ 
(6)       $fin \leftarrow \text{true}$ 
(7)    else
(8)      ( $fin, (c_0, c_1, c_2)$ )  $\leftarrow$  REGIONSUIVANTE( $\mathcal{S}, (c_0, c_1, c_2), x', w$ )
(9)  return ( $i_{\text{obj}}, t_{\text{obj}}, I_{\text{obj}}$ )

```

L'algorithme TRAITERREGION (algorithme 13) retourne l'indice du premier objet rencontré par la rayon considéré et le paramètre correspondant sur ce rayon. Par convention, s'il n'y a pas d'intersection alors le triplet $(-1, +\infty, \{\})$ est renvoyé.

Quatre cas de figure se présentent pour la recherche d'intersection dans une région (figure 3.25). Pour un objet O^i lié à la région R_{k_j} considérée, dont le point d'intersection avec un rayon $R_{x,w}$ est x_{obj} , il faudra s'assurer que x_{obj} est bien dans R_{k_j} .

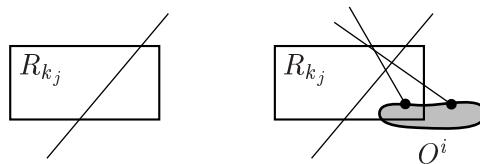


Fig. 3.25 – Différents cas de figure pour la recherche d'intersection dans une région.

Si il existe, le point d'intersection recherché x_{obj} vérifie :

$$\exists j \in \{1, \dots, m\} \text{ tel que } \begin{cases} \exists i \in L_{R_{k_j}} \text{ tel que } x_{\text{obj}} \in R_{x,w} \cap O^i \\ x_{\text{obj}} \in R_{k_j} \end{cases} .$$

L'algorithme TRAITERREGION s'écrit de la manière suivante.

Algorithme 13: *Recherche d'une intersection dans la région courante.*

```

TRAITERREGION( $\mathcal{S}, (c_0, c_1, c_2), x, w$ )
(1)  ( $i_{\text{obj}}, t_{\text{obj}}, I_{\text{obj}}$ )  $\leftarrow (-1, +\infty, \{\})$ 
(2)  if  $L_{(c_0, c_1, c_2)} \neq \emptyset$ 
(3)    for  $j = 1$  to  $\text{taille}(L_{(c_0, c_1, c_2)})$ 
(4)       $i \leftarrow L_{(c_0, c_1, c_2)}(j)$ 
(5)      ( $t, I$ )  $\leftarrow$  INTERRAYONOBJET( $x, w, O^i$ )
(6)      if  $t < t_{\text{obj}}$ 
(7)        ( $i_{\text{obj}}, t_{\text{obj}}, I_{\text{obj}}$ )  $\leftarrow (i, t, I)$ 
(8)  if  $t_{\text{obj}} < +\infty$ 
(9)     $x_{\text{obj}} \leftarrow x + t_{\text{obj}} w$ 
(10)   if  $x_{\text{obj}} \notin R_{(c_0, c_1, c_2)}$ 
(11)     ( $i_{\text{obj}}, t_{\text{obj}}, I_{\text{obj}}$ )  $\leftarrow (-1, +\infty, \{\})$ 
(12)  return ( $i_{\text{obj}}, t_{\text{obj}}, I_{\text{obj}}$ )

```

L'algorithme `INTERRAYONOBJET` recherche la première intersection entre un rayon et un objet. Si une intersection existe, alors il renvoie le paramètre correspondant sur le rayon et, éventuellement, diverses informations relatives à l'objet et au point d'intersection trouvé (vecteur normal au point trouvé, point dans l'espace des paramètres pour un objet paramétrique, numéro de la face pour un objet BRep, etc). Sinon, par convention, il renvoie $(+\infty, \{\})$.

Pour accélérer le parcours du rayon dans les espaces vides, deux améliorations sont proposées (figure 3.26). La première consiste à tenir compte d'une boîte englobante pour la recherche d'intersection avec chaque objet, de manière similaire à [80]. La seconde consiste à éviter les calculs redondants avec un même objet, de manière similaire à [1].

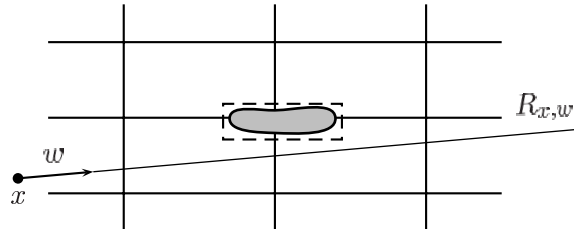


Fig. 3.26 – Accélération du parcours du rayon.

3.2.2 Hiérarchisation

Définir une hiérarchie de subdivisions uniformes c'est définir une partition de régions uniformes du domaine X considéré, puis répéter cette étape aux régions, et ainsi de suite jusqu'à un certain critère d'arrêt (figure 3.27). Ce type de partition peut être vue comme une généralisation de la partition octale.

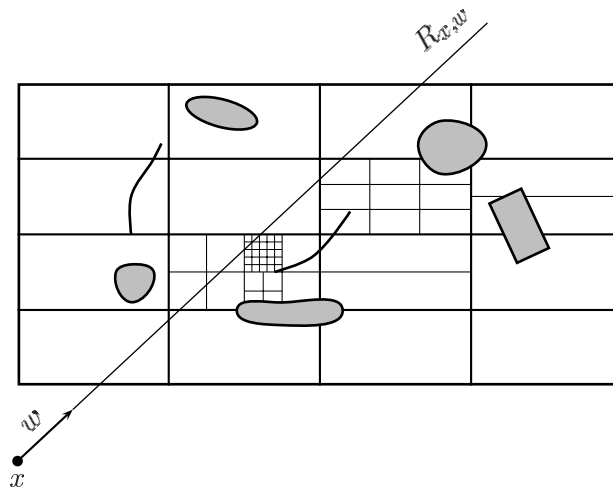


Fig. 3.27 – Hiérarchie de subdivisions uniformes.

De manière analogue à la méthode de subdivision uniforme, cette méthode comporte deux phases :

- une phase de pré-calcul consistant à localiser les objets dans la hiérarchie,
- une phase de calcul consistant à définir le parcours d'un rayon dans la hiérarchie.

Localisation des objets

De même qu'il est nécessaire de repérer les régions dans une subdivision uniforme, il est nécessaire de repérer les subdivisions dans une hiérarchie. Pour ce faire, introduisons une convention de numérotation. Tout d'abord, la numérotation des régions dans une subdivision uniforme est celle introduite

au paragraphe précédent. Ensuite, la subdivision du domaine X est la subdivision mère dans la hiérarchie, elle sera notée \mathcal{S} . Puis, une subdivision d'une région sera notée \mathcal{S}_α , où α est un multi-indice qui contient toute l'information relative à l'arborescence. Par exemple, sur la figure 3.27, la subdivision de la région 5 de \mathcal{S} est noté \mathcal{S}_5 et la subdivision de la région 2 de \mathcal{S}_5 est notée $\mathcal{S}_{5,2}$ (figure 3.28).

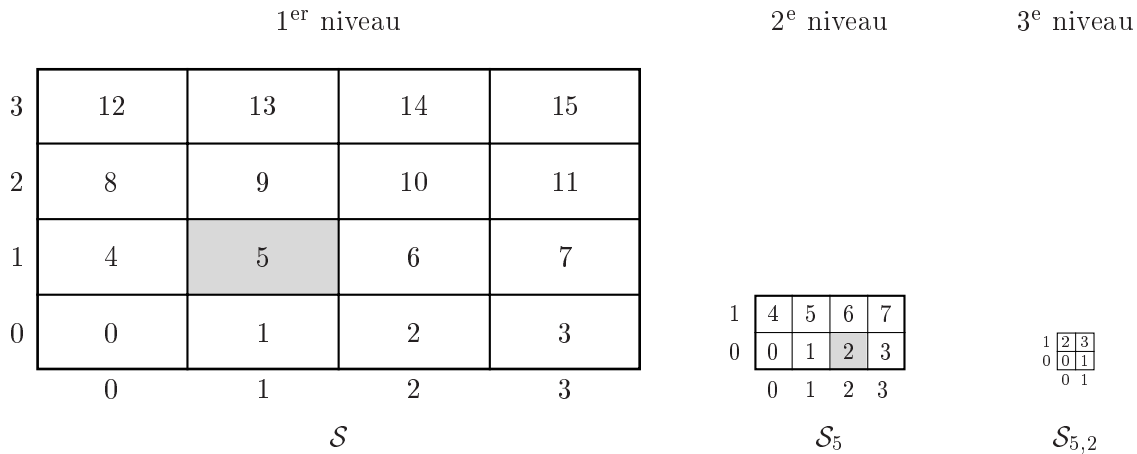


Fig. 3.28 – Convention de numérotation dans une hiérarchie de subdivisions uniformes.

Plus généralement, une subdivision uniforme \mathcal{S}_α est définie de la manière suivante :

$$\mathcal{S}_\alpha = \{R_{\alpha,k}\}_{k=1}^{n_\alpha} ,$$

où n_α est le nombre de régions de la subdivision uniforme considérée. On obtient ainsi une représentation arborescente pour une hiérarchie de subdivisions uniformes (figure 3.29).

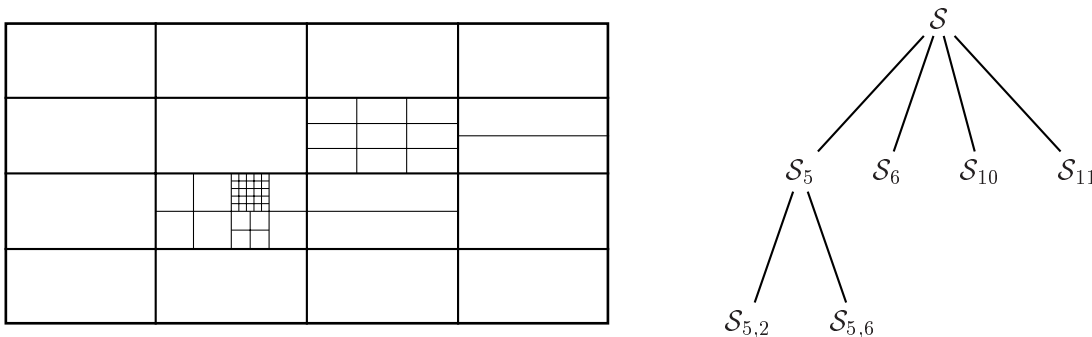


Fig. 3.29 – Représentation arborescente d'une hiérarchie de subdivisions uniformes.

La localisation des objets dans une hiérarchie consiste à localiser les objets dans chaque subdivision uniforme puis à placer ces subdivisions dans une structure arborescente gérant les liens entre les différentes subdivisions. Ainsi, des listes $L_{R_{\alpha,k}}$ sont associées à chaque région $R_{\alpha,k}$ de la hiérarchie :

$$L_{R_{\alpha,k}} = \{i, \Upsilon(O^i) \cap R_{\alpha,k} \neq \emptyset\} ,$$

où, comme dans le paragraphe précédent, l'intersection entre $\Upsilon(O^i)$ et $R_{\alpha,k}$ dépend du type de localisation des objets choisie. Une structure de données doit être mise en place pour le stockage en mémoire des liens entre les objets et les régions des différentes subdivisions de la hiérarchie (figure 3.30). Une

structure de données de type arborescente, bien que naturelle, n'est pas adaptée pour notre contexte. D'une part, les informations reliant les différentes subdivisions et les régions associées demanderaient la gestion d'une très grande quantité de pointeurs. D'autre part, cette représentation induirait un morcellement de la mémoire, qui pourrait dégrader les coûts de parcours de ce type de structure. Il est par conséquent nécessaire de définir une structure de données en allouant le minimum de blocs mémoires et dont la gestion est relativement simple, de manière à obtenir des coûts faibles de parcours de cette structure. Une structure de données comportant plusieurs listes, analogues à celles décrites dans le paragraphe précédent (figure 3.20), est bien adaptée.

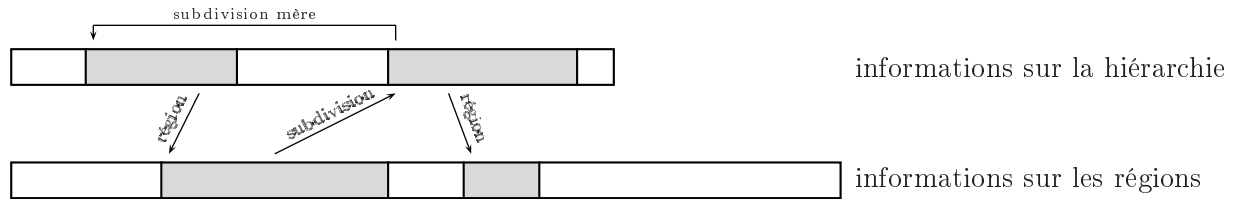


Fig. 3.30 – Structure de données pour l'enregistrement des liens entre les objets et les régions dans une hiérarchie de subdivisions uniformes.

Parcours du rayon

Le parcours d'un rayon dans une hiérarchie de subdivisions uniformes consiste à parcourir en profondeur l'arbre des subdivisions uniformes en parcourant chaque subdivision uniforme rencontrée à l'aide de la méthode décrite dans le paragraphe précédent (figure 3.31). Lors du parcours d'une subdivision uniforme, pour chaque région rencontrée deux cas se présentent. Soit la région est subdivisée, auquel cas il faut parcourir la subdivision de cette région. Soit la région n'est pas subdivisée, auquel cas il faut rechercher une éventuelle intersection avec les objets liés à cette région. Il ne faut pas oublier de traiter le cas où le rayon sort d'une région subdivisée, pour lequel il faut remonter dans l'arbre des subdivisions uniformes.

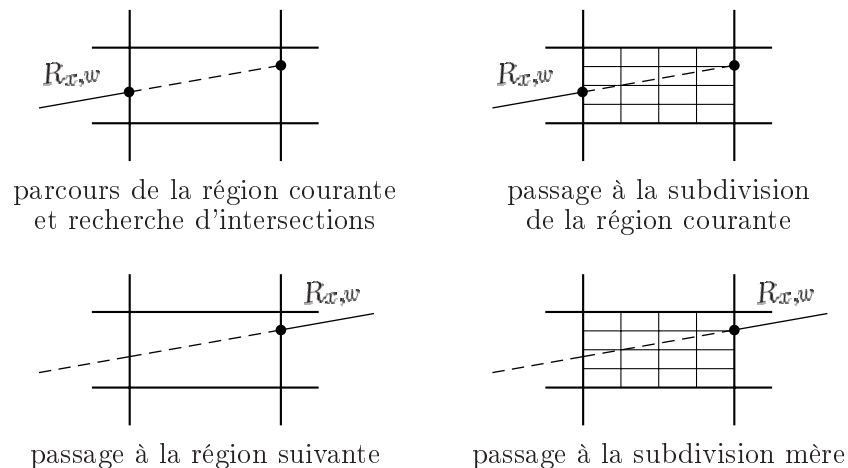


Fig. 3.31 – Parcours du rayon dans une hiérarchie de subdivisions uniformes.

Il est nécessaire de conserver l'information relative au parcours en profondeur, c'est-à-dire de connaître à chaque instant la position dans l'arbre des subdivisions uniformes. C'est le rôle de la notation multi-indicielle. La méthode se décompose en un algorithme de parcours de la hiérarchie (algorithme 14) et en un algorithme de parcours d'une subdivision uniforme (algorithme 15). Chaque région rencontrée qui n'est pas subdivisée est traitée à l'aide de l'algorithme TRAITERREGION (algorithme 13).

Algorithme 14: *Parcours d'un rayon dans une hiérarchie de subdivisions uniformes.*

```

PARCOURSHIERARCHIE( $\mathcal{S}, x, w$ )
(1)  ( $\alpha, suiv, fin$ )  $\leftarrow$  ( $null, false, false$ )
(2)  while not  $fin$ 
(3)    ( $type, (c_0, c_1, c_2), (i_{obj}, t_{obj}, I_{obj})$ )  $\leftarrow$  PARCOURSSUBREGION( $\mathcal{S}_\alpha, (c_0, c_1, c_2), x, w, suiv$ )
(4)    if  $type = sub$ 
(5)       $\alpha \leftarrow$  AJOUTER( $\alpha, INDICE(\mathcal{S}_\alpha, (c_0, c_1, c_2))$ )
(6)       $suiv \leftarrow false$ 
(7)    else if  $type = inter$ 
(8)       $fin \leftarrow true$ 
(9)    else
(10)     if  $\dim(\alpha) = 0$ 
(11)        $fin \leftarrow true$ 
(12)     else
(13)        $k \leftarrow$  DERNIERINDICE( $\alpha$ )
(14)        $\alpha \leftarrow$  ENLEVERDERNIER( $\alpha$ )
(15)        $(c_0, c_1, c_2) \leftarrow$  COORDONNEES( $\mathcal{S}_\alpha, k$ )
(16)        $suiv \leftarrow true$ 
(17)  return ( $i_{obj}, t_{obj}, I_{obj}$ )

```

Si la variable $suiv$ vaut **false** alors le parcours commence dans la région (c_0, c_1, c_2) , sinon $suiv$ vaut **true** et le parcours commence dans la région suivante. Ce paramètre est utile lorsqu'on passe à un niveau supérieur de la hiérarchie. La variable $type$ décrit le type de région courante. Soit elle est subdivisée (sub), soit elle contient une intersection ($inter$) soit c'est une région vide ($vide$). La fonction AJOUTER ajoute à un multi-indice un indice donné. La fonction ENLEVERDERNIER supprime le dernier indice d'un multi-indice donné. Ces deux fonctions permettent de définir les multi-indices pour les passages à un niveau inférieur ou supérieur dans la hiérarchie. La fonction DERNIERINDICE donne le dernier indice d'un multi-indice donné. Les fonctions INDICE et COORDONNEES permettent de passer du repérage d'une région par les coordonnées (c_0, c_1, c_2) au repérage par un indice k et vice-versa.

Algorithme 15: *Parcours d'un rayon dans la subdivision d'une région.*

```

PARCOURSSUBREGION( $\mathcal{S}_\alpha, (c_0, c_1, c_2), x, w, suiv$ )
(1)   $type \leftarrow vide$ 
(2)  if not  $suiv$ 
(3)    ( $fin, (c_0, c_1, c_2), x'$ )  $\leftarrow$  PREMIEREREGION( $\mathcal{S}_\alpha, x, w$ )
(4)  else
(5)     $fin \leftarrow false$ 
(6)  ( $i_{obj}, t_{obj}, I_{obj}$ )  $\leftarrow$   $(-1, +\infty, \{\})$ 
(7)  while not  $fin$ 
(8)    if not  $suiv$ 
(9)      if ESTSUBDIVISE( $\mathcal{S}_\alpha, (c_0, c_1, c_2)$ )
(10)     ( $type, fin$ )  $\leftarrow$  ( $sub, true$ )
(11)    else
(12)     ( $i_{obj}, t_{obj}, I_{obj}$ )  $\leftarrow$  TRAITERREGION( $\mathcal{S}_\alpha, (c_0, c_1, c_2), x, w$ )
(13)     if  $t_{obj} < +\infty$ 
(14)       ( $type, fin$ )  $\leftarrow$  ( $inter, true$ )
(15)    else
(16)      $suiv \leftarrow false$ 
(17)    if not  $fin$ 
(18)     ( $fin, (c_0, c_1, c_2)$ )  $\leftarrow$  REGIONSUIVANTE( $\mathcal{S}_\alpha, (c_0, c_1, c_2), x', w$ )
(19)  return ( $type, (c_0, c_1, c_2), (i_{obj}, t_{obj}, I_{obj})$ )

```

L'algorithme ESTSUBDIVISE teste si une région d'une subdivision uniforme donnée est subdivisée. Les algorithmes PREMIEREREGION, TRAITERREGION et REGIONSUIVANTE sont liés au parcours d'un rayon dans une subdivision uniforme. Il sont décrits dans le paragraphe précédent.

3.3 Résultats expérimentaux

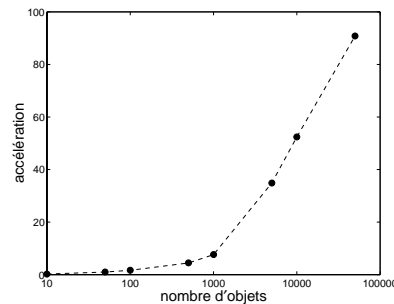
Le développement d'un outil d'accélération des calculs dans le contexte industriel de l'optique s'est déroulé en deux phases. Dans un premier temps, un prototype a été mis en place pour évaluer l'efficacité de la méthode envisagée. Dans un second temps, une fois le prototype validé, l'intégration d'un outil dans l'environnement logiciel a été effectuée. Le paragraphe 3.3.1 présente les résultats obtenus pour le prototype. Le paragraphe 3.3.2 présente les jeux de tests effectués dans l'environnement logiciel, nommé Speos, ainsi que les résultats issus de ces jeux de tests.

3.3.1 Prototype

Le développement d'un prototype s'est effectué dans un environnement de test, distinct de l'environnement logiciel, possédant cependant en commun quelques bibliothèques. L'objectif de ce développement est de mettre en évidence et quantifier l'efficacité de la méthode proposée dans le paragraphe 3.2 par rapport au nombre d'objets considéré.

Une des restrictions de cet environnement de test est de n'offrir qu'une partie du modèleur géométrique du logiciel final, à savoir les objets primitifs CSG, sans disposer des opérateurs ensemblistes. Les scènes testées sont donc composées d'objets primitifs (section 2.2). La répartition des objets est quelconque, et leur dimension, caractérisée par les paramètres propres à l'objet primitif considéré, est quelconque. On s'intéresse principalement à faire varier le nombre d'objets primitifs.

Pour quantifier l'efficacité de la méthode de localisation par subdivision spatiale, une comparaison du temps d'exécution est effectuée pour chaque scène testée avec la méthode alors utilisée dans l'environnement de développement. C'est une méthode de type naïf, les calculs d'intersection rayon-objet sont accélérés à l'aide de sphères englobantes. Les résultats sont présentés dans le graphe suivant, où l'axe des abscisses représente le nombre d'objets primitifs de la scène testée et l'axe des ordonnées représente l'accélération. Dans le contexte industriel, la diversité des systèmes optiques étudiés montre que le nombre d'objets dans les scènes géométriques va de quelques dizaines à plusieurs milliers.



Le graphe précédent donne un idée de l'efficacité de la nouvelle méthode. En effet, pour les objets considérés dans ces tests, l'efficacité de la méthode de localisation par subdivision spatiale devient intéressante à partir de 1 000 objets où l'accélération est de l'ordre de 10. Pour un nombre d'objets encore plus grand la méthode devient encore plus efficace, pour atteindre un facteur d'accélération de l'ordre de 90 pour 50 000 objets. On remarque que la courbe de l'accélération est linéaire, puisque l'échelle des abscisses du graphe est logarithmique, l'accélération est donc logarithmique, par rapport au nombre d'objets.

Ce constat est satisfaisant pour l'entreprise, on peut envisager le développement de la méthode dans l'environnement logiciel.

3.3.2 Environnement logiciel

Le développement de l'outil d'accélération envisagé nécessite un travail plus important dans l'environnement logiciel que dans l'environnement de test. D'une part, pour des considérations de qualité, la bibliothèque doit être robuste, ce qui implique une méthode de développement particulière. D'autre part, le modeleur géométrique hybride est à considérer. Par conséquent, les scènes géométriques peuvent contenir des objets CSG et des objets BRep. L'évolution du logiciel Speos a impliqué plusieurs développements d'une bibliothèque d'accélération. Nous présentons ici des jeux de tests effectués sur la dernière version.

La bibliothèque qui a été développée peut être paramétrée manuellement. Mais, pour un utilisateur du logiciel Speos, les deux phases du calcul (pré-calcul et simulation) doivent être transparentes, et, surtout, ne pas nécessiter son intervention. Pour cela, il est nécessaire de fixer certains paramètres a priori. Tout d'abord, pour connaître, en particulier, de manière explicite la frontière des objets volumiques, les objets CSG sont convertis en objets BRep. Pour ce faire, chaque objet primitif d'un objet CSG est associé à un objet BRep. Les différents opérateurs géométriques et ensemblistes sont pris en compte au cours de cette conversion. Les deux modèles géométriques sont conservés en mémoire. À chaque objet BRep est associé une boîte englobante qui sera utilisée pour la localisation. Pour les calculs d'intersection rayon-objet, le modèle géométrique CSG est utilisé pour les objets CSG. Ensuite, pour la phase de pré-calcul, c'est-à-dire la mise en place d'une hiérarchie de subdivisions uniformes, les critères suivants sont considérés. Soit le nombre de subdivisions dans chaque direction est fixé, soit, pour la direction i , c'est le nombre donné dans [59] :

$$n_i = \left\lceil \left(\frac{\Delta_i^2 N}{\Delta_{i+1} \Delta_{i+2}} \right)^{1/3} \right\rceil ,$$

où Δ_i est la longueur dans la direction i du parallélépipède à subdiviser, Δ_{i+1} et Δ_{i+2} doivent être compris comme $\Delta_{(i+1) \bmod 3}$ et $\Delta_{(i+2) \bmod 3}$, N le nombre d'objets dans le voxel considéré et $\lceil \cdot \rceil$ l'opérateur numérique d'arrondi (valant 1 au minimum pour être cohérent avec un nombre de subdivisions). Le critère d'arrêt est une profondeur de la hiérarchie maximale (5 niveaux) et un seuil pour le nombre maximum d'objets par voxel (5 objets).

Les simulations sont effectués à l'aide d'un algorithme de type Monte-Carlo. Des photons sont émis depuis les sources et sont propagés dans le système jusqu'à atteindre une carte réceptrice ou être absorbés. L'intervention de l'utilisateur se limite à la définition des caractéristiques de la simulation, à savoir le nombre de photons à émettre et le nombre de photons enregistrés sur les cartes réceptrices. Il est nécessaire de préciser que les jeux de tests n'ont pu être effectués que sur des scènes contenant un nombre très limité d'objets (environ 2 000 objets au maximum), et ce pour des raisons de coût mémoire du modeleur géométrique hybride (c'est essentiellement la partie BRep qui est pénalisante).

Les jeux de tests sont caractérisés, d'une part, par le système optique, et notamment par sa géométrie, et, d'autre part, par des paramètres observés pendant les différentes simulations envisagées. Les deux paragraphes suivants décrivent les différents paramètres observés. Le troisième paragraphe présente les résultats des tests.

Paramètres géométriques d'une scène

Pour caractériser une scène géométrique, on rend compte des paramètres suivants :

- répartition des types d'objets,
- répartition de la dimension des objets (longueur de la diagonale de la boîte englobante),
- répartition de l'aire de la surface optique des objets.

La présentation de ces paramètres est la suivante. La répartition des types de surface des objets (après conversion vers le modèle BRep) est donnée dans un tableau. La répartition de la dimension

des objets est donnée par un graphique comprenant l’histogramme des dimensions relatives (rapport entre la dimension de l’objet et la dimension maximale) et la courbe de la dimension relative cumulée. De manière similaire, la répartition de l’aire des objets est donnée par un graphique comprenant l’histogramme des aires relatives (rapport entre l’aire de la surface optique de l’objet et l’aire maximale) et la courbe de l’aire relative cumulée.

Paramètres observés lors d’une simulation

Trois types de simulations sont testées pour chaque système optique envisagé :

- simulation avec la méthode naïve (algorithme 2 pour les objets CSG et algorithme 5 pour les objets BRep), c’est-à-dire sans localisation (type SL),
- simulations avec localisation dans une subdivision uniforme (type LU, de 1 à 32 subdivisions dans chaque direction),
- simulation avec localisation dans une hiérarchie de subdivisions uniformes (type LHU).

Différents paramètres sont observés pendant la simulation. Il sont présentés dans un tableau, dont la définition dans l’ordre des lignes est la suivante.

- Le temps de pré-calcul est le temps nécessaire pour lire et éventuellement convertir des objets dans le modèle BRep et pour construire la structure de données définissant la localisation.
- Le temps de simulation représente uniquement le temps total des calculs optiques (émission des photons, calculs d’intersection et d’interaction photon-objet, enregistrement sur des cartes d’observation), le pré-calcul n’est pas pris en compte.
- La taille de la structure de données correspond à la quantité de mémoire nécessaire pour stocker la structure de données relative aux informations de la localisation (liens voxel-objet).
- Le nombre moyen d’objets et le pourcentage de voxels vides par voxel sont des paramètres donnés dans le cas des simulations de type SU.
- Le nombre de photons émis correspond au nombre total de photons émis par les objets sources. C’est un paramètre réglé avant la simulation.
- Le nombre de rayons traités correspond au nombre de rayons qui sont engendrés par le parcours des photons. Un photon émis engendre au moins un rayon. À chaque interaction avec un objet, soit il y a absorption, soit un nouveau rayon est à traiter.
- Le nombre d’intersections correspond au nombre total d’intersections recherchées pour l’ensemble des rayons traités. Parmi ces intersections, on différencie les intersections réellement calculées (calcul d’intersection rayon-surface) et les intersections non calculées (dans le cas de recherches répétées d’intersections entre un même rayon et un même objet, le calcul n’est effectué qu’une seule fois).
- Le nombre d’échecs rend compte du nombre de recherches d’intersections qui n’aboutissent pas (soit il n’y a pas intersection, soit l’intersection trouvée n’est pas le point recherché). Pour un rayon donné, le nombre d’échecs est donc le nombre de calculs d’intersection entre ce rayon et des objets auquel on enlève un s’il y a bien existence d’une intersection avec le rayon considéré.
- Le nombre moyen de voxels traversés par rayon est la somme des nombres de voxels traversés pour chaque rayon divisée par le nombre de rayons.
- La longueur moyenne d’un rayon est la somme des longueurs de chaque rayon divisée par le nombre de rayons.
- Le nombre moyen d’intersections par rayon est la somme des nombres d’intersections réellement calculées pour chaque rayon divisée par le nombre de rayons.

- Le pourcentage d'intersections calculées est le rapport du nombre d'intersections calculées sur le nombre d'intersections.
- Le pourcentage d'échecs est le rapport du nombre d'échecs moins le nombre d'intersections non calculées sur le nombre d'intersections.
- Le temps moyen de simulation par rayon est le rapport du temps de simulation sur le nombre de rayons traités.
- Le temps moyen d'intersection par rayon-objet est la somme des temps cumulés de recherche d'intersections pour chaque objet divisée par le nombre d'intersections recherchées.
- Le temps moyen de simulation par rayon-objet est le temps de simulation divisé par le nombre d'intersections recherchées.

Deux graphiques sont tirés de ce tableau. Le premier rend compte de quantités moyennes par rayon (nombre moyen d'intersections par rayon et temps moyen de simulation par rayon). Le second rend compte de quantités moyennes par rayon-objet (temps moyen d'intersection par rayon et par objet et temps moyen de simulation par rayon et par objet).

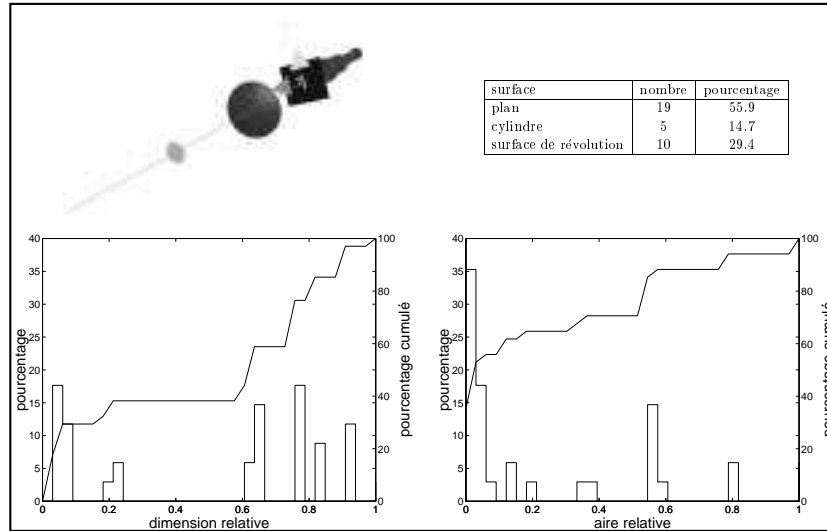
Enfin, un tableau présente les accélérations obtenues pour la simulation de type LHU par rapport à la simulation de type SL et par rapport à la simulation de type LU (1,1,1). L'intérêt de différencier les deux accélérations est principalement de prendre en compte les scènes où la géométrie considérée dans la simulation de type SL est différente de la géométrie considérée pour la localisation dans les simulations de type LU et LHU.

Résultats

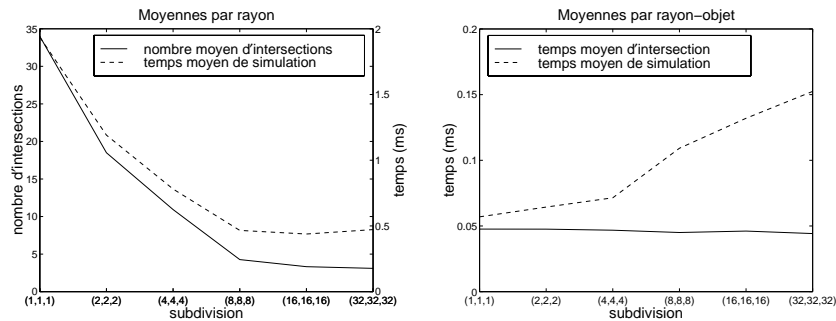
Chaque test est présenté sur une page séparée en deux parties (caractéristiques géométriques et simulations). Les systèmes optiques testés sont les suivants :

- (1) alignement de lentilles,
- (2) aiguille de tableau de bord,
- (3) compteur d'automobile,
- (4) ampoule,
- (5) matrice de lentilles,
- (6) matrice d'objets,
- (7) système optique,
- (8) système d'éclairage d'intérieur.

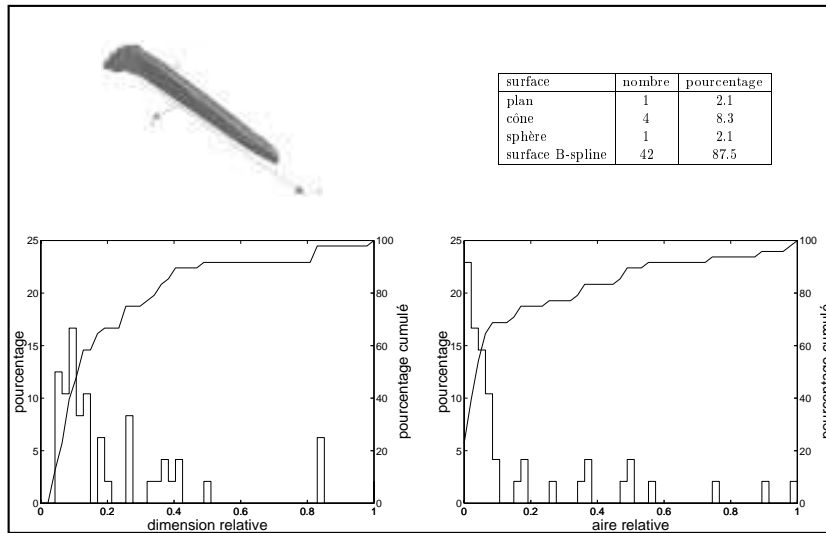
(1) Alignement de lentilles



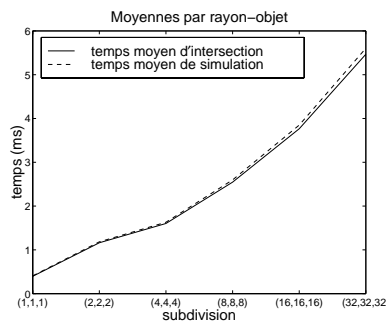
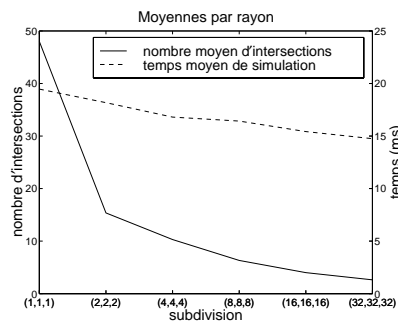
paramètre	SL	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)	LHU
temps de pré-calcul (s)		3	3	3	3	6	17	2
temps de simulation (s)	196	652	401	263	153	143	153	204
taille de la structure de données (octets)		240	1 560	7 624	32 800	108 996	372 216	5 580
nombre moyen d'objets par voxel		34.00	13.25	4.96	2.49	2.14	1.98	
pourcentage de voxels vides		0.00	0.00	12.50	47.85	77.98	90.53	
nombre de photons émis	100 005	100 025	100 019	100 023	100 033	100 020	100 000	100 001
nombre de rayons traités	337 351	336 705	336 780	336 963	328 524	326 576	323 857	335 769
nombre d'intersections	3 373 510	11 447 970	8 083 861	5 024 307	2 300 043	1 726 046	1 891 498	3 358 092
nombre d'intersections calculées	3 373 510	11 447 970	6 228 648	3 682 942	1 402 030	1 084 453	1 002 760	2 298 761
nombre d'intersections non calculées	0	0	1 855 213	1 341 365	898 013	641 593	888 738	1 059 331
nombre d'échecs	3 072 459	11 147 465	7 783 471	4 723 434	2 007 989	1 436 330	1 604 001	3 058 633
nombre moyen de voxels traversés par rayon		1.00	2.11	2.84	5.32	8.11	13.12	4.52
longueur moyenne d'un rayon		48.27	48.22	48.27	49.75	50.13	50.48	48.23
nombre moyen d'intersections par rayon	10.00	34.00	18.49	10.93	4.27	3.32	3.10	6.85
pourcentage d'intersections calculées	100.00	100.00	77.05	73.30	60.96	62.83	53.01	68.45
pourcentage d'échecs	91.08	97.38	73.33	67.31	48.26	46.04	37.81	59.54
temps moyen de simulation par rayon (ms)	0.58	1.94	1.19	0.78	0.47	0.44	0.47	0.61
temps moyen d'intersection rayon-objet (ms)		0.05	0.05	0.05	0.05	0.05	0.04	0.05
temps moyen de simulation rayon-objet (ms)	0.06	0.06	0.06	0.07	0.11	0.13	0.15	0.09



(2) Aiguille de tableau de bord

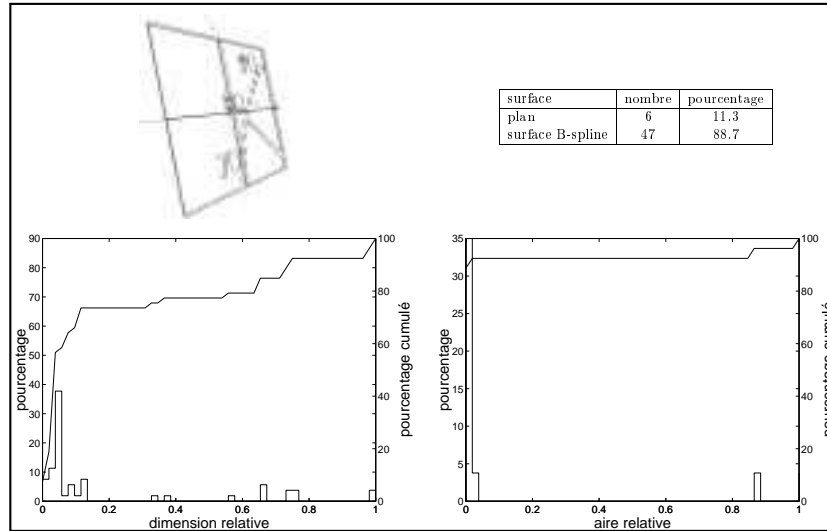


paramètre	SL	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)	LHU
temps de pré-calcul (s)		4	4	4	5	9	39	3
temps de simulation (s)	846	847	807	742	724	680	649	734
taille de la structure de données (octets)		296	1 572	4 312	16 452	77 944	325 536	20 892
nombre moyen d'objets par voxel		48.00	11.88	5.47	2.83	1.73	1.34	
pourcentage de voxels vides		0.00	0.00	53.13	74.41	84.06	91.54	
nombre de photons émis	5 000	5 000	5 000	5 000	5 000	5 000	5 000	5 000
nombre de rayons traités	44 701	43 531	44 398	44 178	44 100	44 097	44 001	44 548
nombre d'intersections	2 145 648	2 089 488	823 950	600 650	430 328	313 388	259 711	655 994
nombre d'intersections calculées	2 145 648	2 089 488	682 067	455 038	278 863	176 655	115 817	400 238
nombre d'intersections non calculées	0	0	141 883	145 612	151 465	136 733	143 894	255 756
nombre d'échecs	2 105 947	2 050 957	784 552	561 472	391 228	274 291	220 710	616 446
nombre moyen de voxels traversés par rayon		1.00	2.63	3.74	6.48	10.00	16.21	7.74
longueur moyenne d'un rayon		5.34	5.27	5.31	5.32	5.33	5.33	5.28
nombre moyen d'intersections par rayon	48.00	48.00	15.36	10.30	6.32	4.01	2.63	8.98
pourcentage d'intersections calculées	100.00	100.00	82.78	75.76	64.80	56.37	44.59	61.01
pourcentage d'échecs	98.15	98.16	78.00	69.23	55.72	43.89	29.58	54.98
temps moyen de simulation par rayon (ms)	18.92	19.47	18.18	16.80	16.43	15.42	14.75	16.48
temps moyen d'intersection rayon-objet (ms)		0.40	1.16	1.60	2.55	3.77	5.47	1.79
temps moyen de simulation rayon-objet (ms)	0.39	0.41	1.18	1.63	2.60	3.85	5.60	1.83

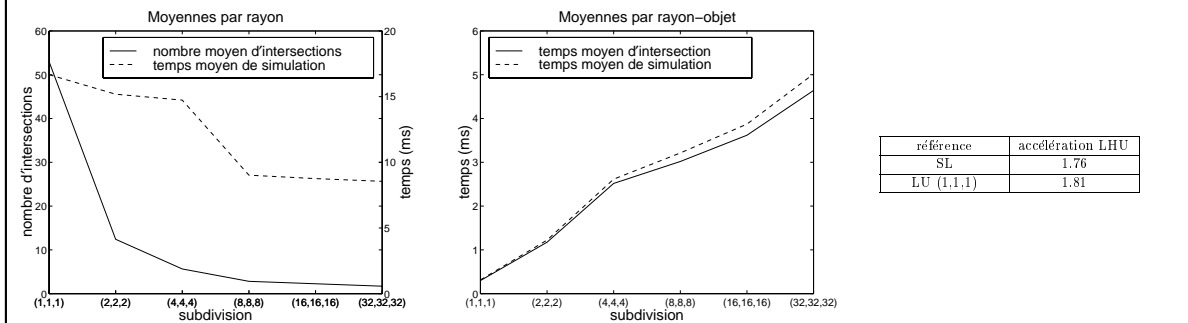


référence	accélération LHU
SL	1.15
LU (1,1,1)	1.18

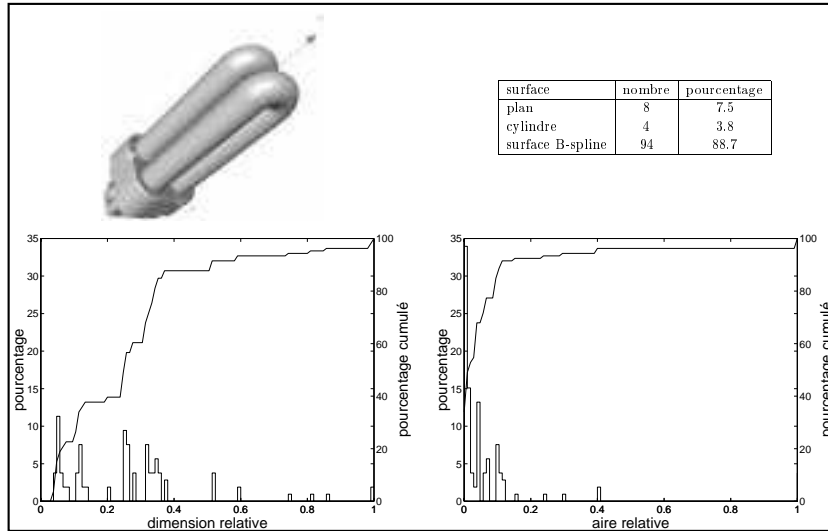
(3) Compteur d'automobile



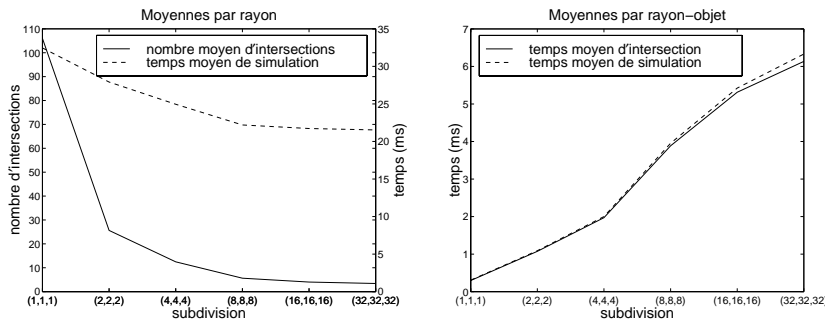
paramètre	SL	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)	LHU
temps de pré-calcul (s)		2	2	2	4	16	107	1
temps de simulation (s)	814	834	760	737	451	438	428	461
taille de la structure de données (octets)		316	1 608	8 684	45 940	213 052	1 234 356	34 560
nombre moyen d'objets par voxel		53.00	12.38	4.69	2.34	1.66	1.32	
pourcentage de voxels vides		0.00	0.00	0.00	26.56	56.23	67.89	
nombre de photons émis	50 001	50 003	50 016	50 001	50 009	50 010	50 009	50 011
nombre de rayons traités	50 001	50 003	50 016	50 001	50 009	50 010	50 009	50 011
nombre d'intersections	2 400 048	2 650 159	626 086	309 953	173 391	177 888	207 031	271 273
nombre d'intersections calculées	2 400 048	2 650 159	620 797	281 703	140 436	113 094	85 220	204 185
nombre d'intersections non calculées	0	0	5 289	28 250	32 955	64 794	121 811	67 088
nombre d'échecs	2 378 457	2 628 366	604 340	288 332	151 732	156 348	185 372	249 622
nombre moyen de voxels traversés par rayon		1.00	1.64	1.90	2.34	4.06	8.40	3.24
longueur moyenne d'un rayon		0.42	0.42	0.42	0.41	0.41	0.42	0.42
nombre moyen d'intersections par rayon	48.00	53.00	12.41	5.63	2.81	2.26	1.70	4.08
pourcentage d'intersections calculées	100.00	100.00	99.16	90.89	80.99	63.58	41.16	75.27
pourcentage d'échecs	99.10	99.18	95.68	83.91	68.50	51.47	30.70	67.29
temps moyen de simulation par rayon (ms)	16.28	16.67	15.19	14.74	9.02	8.76	8.55	9.22
temps moyen d'intersection rayon-objet (ms)		0.30	1.17	2.52	3.02	3.62	4.64	2.12
temps moyen de simulation rayon-objet (ms)	0.34	0.31	1.22	2.62	3.21	3.87	5.02	2.26



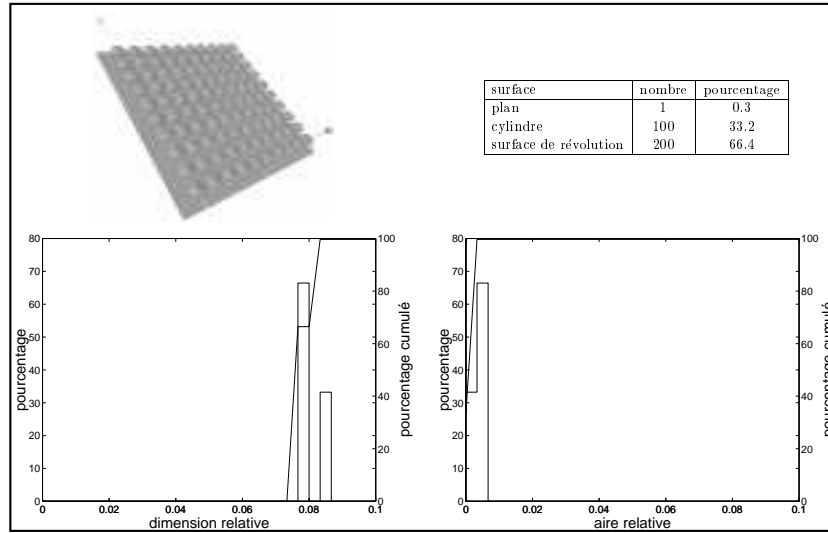
(4) Ampoule



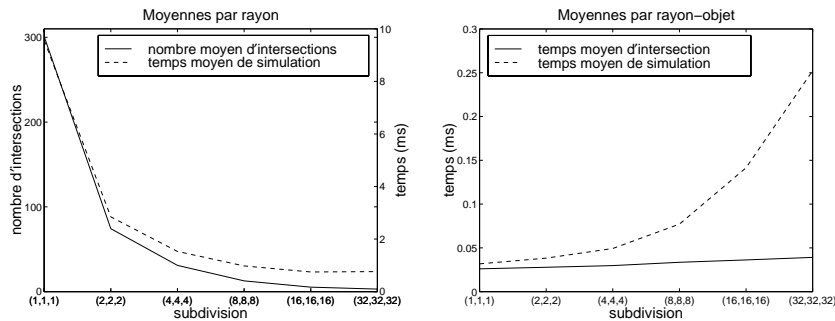
paramètre	SL	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)	LHU
temps de pré-calcul (s)		2	2	2	2	4	17	2
temps de simulation (s)	1 063	1 172	1 032	877	768	757	777	790
taille de la structure de données (octets)		528	2 368	9 808	46 352	310 912	2 202 816	83 504
nombre moyen d'objets par voxel		106.00	29.50	8.25	3.82	2.17	1.79	
pourcentage de voxels vides		0.00	0.00	0.00	29.69	37.21	43.59	
nombre de photons émis	5 003	5 002	5 002	5 002	5 002	5 001	5 004	5 001
nombre de rayons traités	36 229	36 104	36 948	35 150	34 573	34 842	36 060	34 736
nombre d'intersections	3 550 442	3 827 024	1 079 920	630 296	373 599	393 249	600 746	396 324
nombre d'intersections calculées	3 550 442	3 827 024	946 389	438 645	194 077	139 575	122 655	214 570
nombre d'intersections non calculées	0	0	133 531	191 651	179 522	253 674	478 091	181 754
nombre d'échecs	3 519 213	3 795 920	1 047 972	600 146	344 026	363 407	569 686	366 588
nombre moyen de voxels traversés par rayon		1.00	1.77	3.71	6.90	13.50	26.62	8.20
longueur moyenne d'un rayon		11.03	10.78	10.80	10.77	10.85	11.01	10.90
nombre moyen d'intersections par rayon	98.00	106.00	25.61	12.48	5.61	4.01	3.40	6.18
pourcentage d'intersections calculées	100.00	100.00	87.64	69.59	51.95	35.49	20.42	54.14
pourcentage d'échecs	99.12	99.19	84.68	64.81	44.03	27.90	15.25	46.64
temps moyen de simulation par rayon (ms)	29.35	32.46	27.92	24.96	22.20	21.73	21.54	22.73
temps moyen d'intersection rayon-objet (ms)		0.30	1.07	1.97	3.89	5.31	6.14	3.61
temps moyen de simulation rayon-objet (ms)	0.30	0.31	1.09	2.00	3.95	5.42	6.33	3.68



(5) Matrice de lentilles

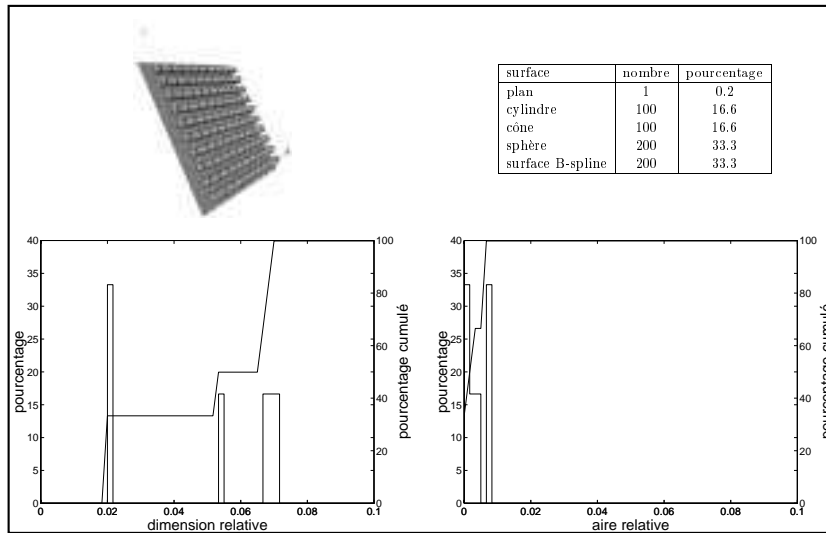


paramètre	SL	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)	LHU
temps de pré-calcul (s)		162	162	162	163	169	209	157
temps de simulation (s)	560	1 887	563	303	194	148	151	184
taille de la structure de données (octets)		1 308	3 420	6 672	23 152	88 232	518 436	341 416
nombre moyen d'objets par voxel		301.00	38.00	13.91	5.92	2.96	1.58	
pourcentage de voxels vides		0.00	0.00	50.00	68.55	82.86	86.66	
nombre de photons émis	100 000	100 000	100 000	100 000	100 000	100 000	100 000	100 000
nombre de rayons traités	197 963	197 282	197 959	198 397	198 591	197 728	198 542	198 168
nombre d'intersections	19 994 263	59 381 882	14 704 818	6 154 043	2 993 405	1 462 898	1 129 927	2 364 994
nombre d'intersections calculées	19 994 263	59 381 882	14 704 656	6 150 070	2 517 186	1 045 512	599 334	1 345 808
nombre d'intersections non calculées	0	0	162	3 973	476 219	417 386	530 593	1 019 186
nombre d'échecs	19 896 300	59 284 600	14 606 859	6 055 646	2 894 814	1 365 170	1 031 385	2 266 826
nombre moyen de voxels traversés par rayon		1.00	2.79	3.84	6.54	11.65	22.37	13.84
longueur moyenne d'un rayon		3.01	3.01	3.01	3.00	3.01	3.01	3.01
nombre moyen d'intersections par rayon	101.00	301.00	74.28	31.00	12.68	5.29	3.02	6.79
pourcentage d'intersections calculées	100.00	100.00	100.00	99.94	84.09	71.47	53.04	56.91
pourcentage d'échecs	99.51	99.84	99.33	98.34	80.80	64.79	44.32	52.75
temps moyen de simulation par rayon (ms)	2.83	9.57	2.85	1.53	0.98	0.75	0.76	0.93
temps moyen d'intersection rayon-objet (ms)		0.03	0.03	0.03	0.03	0.04	0.04	0.04
temps moyen de simulation rayon-objet (ms)	0.03	0.03	0.04	0.05	0.08	0.14	0.25	0.14

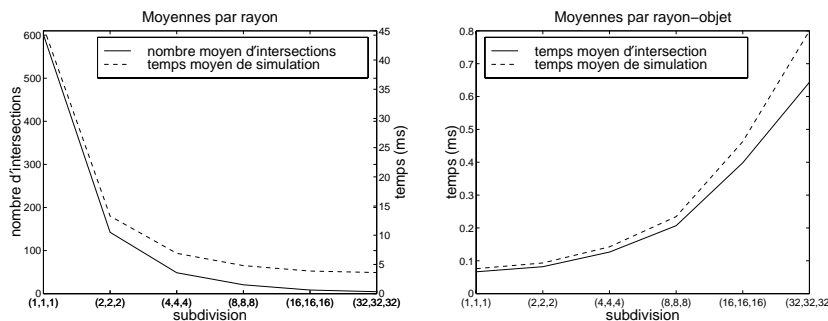


référence	accélération LHU
SL	3.04
LU (1,1,1)	10.29

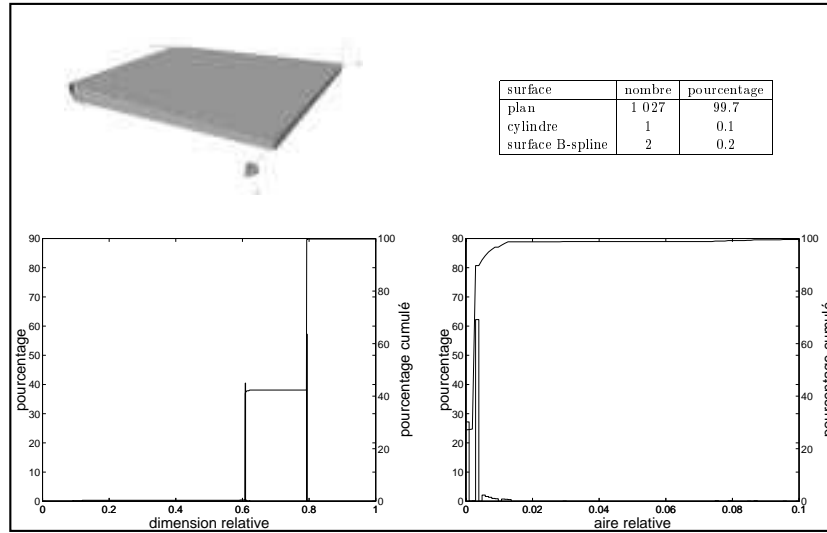
(6) Matrice d'objets



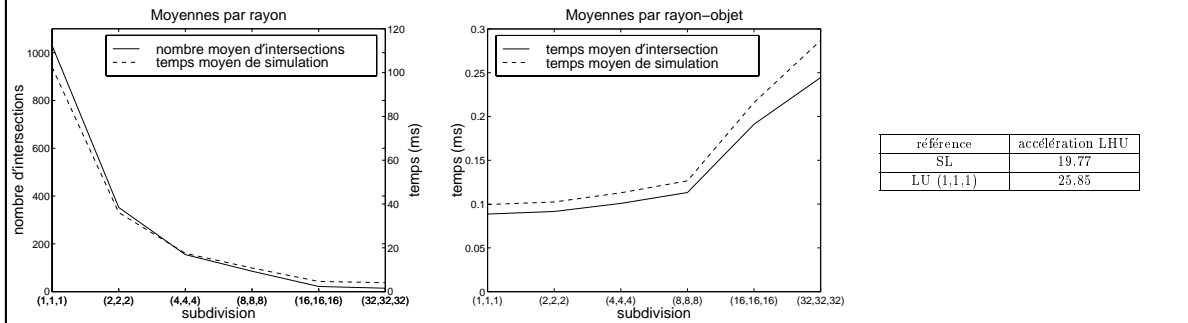
paramètre	SL	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)	LHU
temps de pré-calcul (s)		26	26	26	27	36	108	26
temps de simulation (s)	9 623	10 734	3 120	1 622	1 129	912	850	898
taille de la structure de données (octets)		2 308	6 780	16 152	63 548	301 484	1 825 572	888 024
nombre moyen d'objets par voxel		601.00	105.50	25.30	10.15	5.10	3.06	
pourcentage de voxels vides		0.00	0.00	0.00	21.88	44.87	55.22	
nombre de photons émis	100 000	100 000	100 000	100 000	100 000	100 000	100 000	100 000
nombre de rayons traités	234 780	235 001	234 857	234 835	235 578	235 883	235 621	235 964
nombre d'intersections	141 102 780	141 235 601	38 793 083	17 342 674	10 956 716	7 349 354	6 957 000	6 674 718
nombre d'intersections calculées	141 102 780	141 235 601	33 430 773	11 394 140	4 815 886	1 969 087	1 062 482	1 598 044
nombre d'intersections non calculées	0	0	5 362 310	5 948 534	6 140 830	5 380 267	5 894 518	5 076 674
nombre d'échecs	140 968 000	141 100 600	38 658 226	17 207 839	10 821 138	7 213 471	6 821 379	6 538 754
nombre moyen de voxels traversés par rayon		1.00	2.51	4.96	9.06	16.73	31.55	26.03
longueur moyenne d'un rayon		2.41	2.42	2.42	2.41	2.41	2.42	2.41
nombre moyen d'intersections par rayon	601.00	601.00	142.35	48.52	20.44	8.35	4.51	6.77
pourcentage d'intersections calculées	100.00	100.00	86.18	65.70	43.95	26.79	15.27	23.94
pourcentage d'échecs	99.90	99.90	85.83	64.92	42.72	24.94	13.32	21.90
temps moyen de simulation par rayon (ms)	40.99	45.68	13.28	6.91	4.79	3.86	3.61	3.81
temps moyen d'intersection rayon-objet (ms)		0.07	0.08	0.13	0.21	0.40	0.64	0.46
temps moyen de simulation rayon-objet (ms)	0.07	0.08	0.09	0.14	0.23	0.46	0.80	0.56



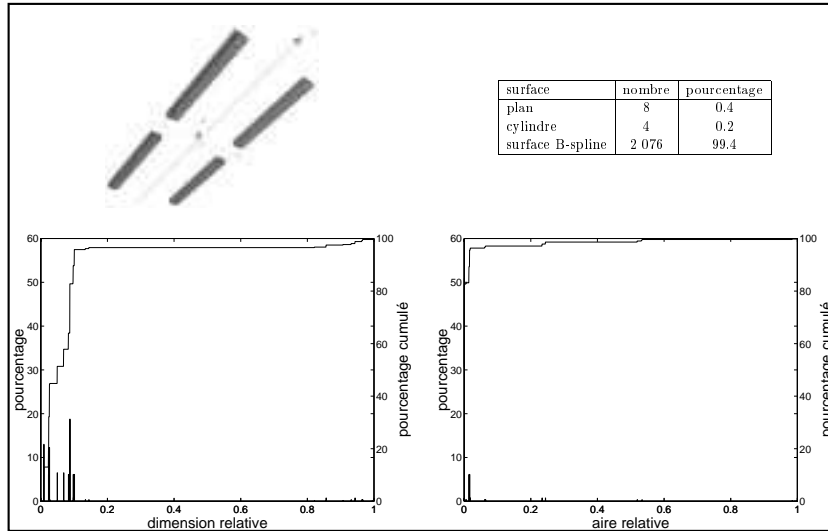
(7) Système optique



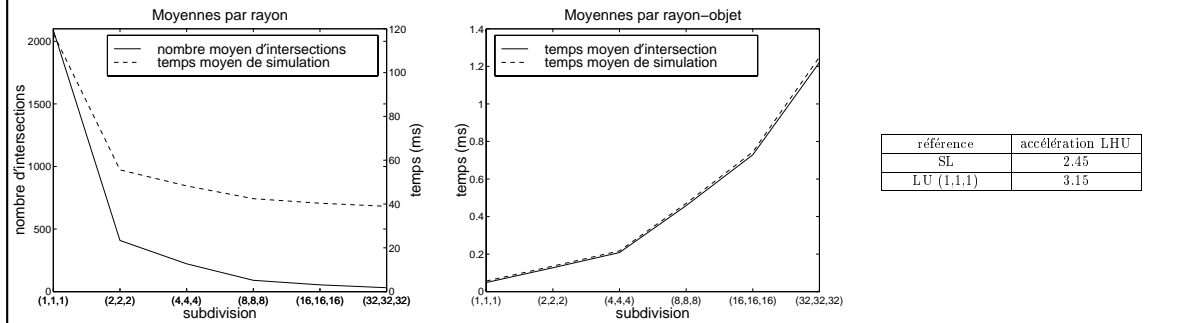
paramètre	SL	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)	LHU
temps de pré-calcul (s)		37	37	37	39	55	177	37
temps de simulation (s)	3 772	4 731	1 656	794	493	216	193	182
taille de la structure de données (octets)		4 224	14 456	31 056	110 064	426 432	2 548 480	34 276 688
nombre moyen d'objets par voxel		1 030.00	291.75	87.66	48.92	19.55	12.16	
pourcentage de voxels vides		0.00	0.00	9.38	32.81	45.80	51.66	
nombre de photons émis	10 019	10 010	10 019	10 019	10 017	10 015	10 019	10 004
nombre de rayons traités	48 120	46 162	45 870	45 446	45 624	46 387	47 141	45 815
nombre d'intersections	49 467 360	47 546 860	17 342 828	8 033 798	4 737 611	1 650 620	1 686 830	1 662 193
nombre d'intersections calculées	49 467 360	47 546 860	16 164 365	7 037 918	3 894 526	999 538	672 937	542 440
nombre d'intersections non calculées	0	0	1 178 463	995 880	843 085	651 082	1 013 893	1 119 753
nombre d'échecs	49 423 220	47 504 568	17 300 878	7 992 192	4 695 907	1 608 163	1 643 639	1 620 368
nombre moyen de voxels traversés par rayon	1.00	2.02	3.91	5.37	8.87	18.35	20.05	
longueur moyenne d'un rayon	1.73	1.76	1.72	1.75	1.73	1.75	1.76	
nombre moyen d'intersections par rayon	1 028.00	1 030.00	352.40	154.86	85.36	21.55	14.27	11.84
pourcentage d'intersections calculées	100.00	100.00	93.20	87.60	82.20	60.56	39.89	32.63
pourcentage d'échecs	99.91	99.91	92.96	87.09	81.32	57.98	37.33	30.12
temps moyen de simulation par rayon (ms)	78.38	102.50	36.09	17.48	10.81	4.65	4.09	3.96
temps moyen d'intersection rayon-objet (ms)	0.08	0.09	0.09	0.10	0.11	0.19	0.24	0.28
temps moyen de simulation rayon-objet (ms)		0.10	0.10	0.11	0.13	0.22	0.29	0.33



(8) Système d'éclairage d'intérieur



paramètre	SL	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)	LHU
temps de pré-calcul (s)		31	31	32	36	68	331	31
temps de simulation (s)	606	829	365	337	274	281	264	253
taille de la structure de données (octets)		8 456	19 496	28 616	87 288	420 360	2 891 608	13 933 120
nombre moyen d'objets par voxel		2 088.00	317.00	50.75	23.32	12.23	7.98	
pourcentage de voxels vides		0.00	0.00	0.00	25.00	37.50	38.87	
nombre de photons émis	1 000	1 000	1 000	1 000	1 000	1 000	1 000	1 000
nombre de rayons traités	6 623	7 037	6 558	6 990	6 460	6 968	6 783	6 755
nombre d'intersections	13 775 840	14 693 256	2 852 176	1 683 470	782 636	698 462	665 005	1 012 964
nombre d'intersections calculées	13 775 840	14 693 256	2 680 956	1 553 642	582 944	377 978	211 392	146 737
nombre d'intersections non calculées	0	0	171 220	129 828	199 692	320 484	453 613	866 227
nombre d'échecs	13 770 217	14 687 219	2 846 618	1 677 480	777 176	692 494	659 222	1 007 209
nombre moyen de voxels traversés par rayon		1.00	1.72	2.18	3.70	7.00	13.87	37.76
longueur moyenne d'un rayon		36.66	42.80	38.37	35.45	40.46	35.91	38.60
nombre moyen d'intersections par rayon	2 080.00	2 088.00	408.81	222.27	90.24	54.24	31.16	21.72
pourcentage d'intersections calculées	100.00	100.00	94.00	92.29	74.48	54.12	31.79	14.49
pourcentage d'échecs	99.96	99.96	93.80	91.93	73.79	53.26	30.92	13.92
temps moyen de simulation par rayon (ms)	91.49	117.81	55.58	48.28	42.44	40.37	38.94	37.41
temps moyen d'intersection rayon-objet (ms)		0.05	0.13	0.21	0.46	0.73	1.22	1.66
temps moyen de simulation rayon-objet (ms)	0.04	0.06	0.14	0.22	0.47	0.74	1.25	1.72



3.3.3 Analyse des résultats

On propose dans ce paragraphe une analyse des expérimentations faites dans l'environnement logiciel. Bien que n'ayant pas pu travailler avec des scènes contenant un grand nombre d'objets, on peut faire des constatations et tirer quelques enseignements des 8 scènes testées. Pour ce faire, on distingue deux types de simulation :

- les simulations basées uniquement sur des subdivisions uniformes (type LU),
- les simulations basées sur une hiérarchie de subdivisions uniformes (type LHU).

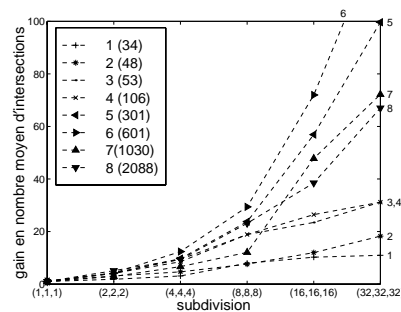
Subdivisions uniformes

Les subdivisions uniformes considérées sont telles que le nombre de subdivisions dans chaque direction est identique. Le test LU (1,1,1) s'apparente au test SL (sans localisation), à la différence près qu'il y a construction d'une structure de données pour la localisation basée sur les objets BRep (issus de la conversion). Les calculs d'intersection rayon-objet sont effectués en utilisant le modèle géométrique original des objets (le modèle CSG étant généralement plus performant). Cependant, lorsqu'un objet CSG se traduit en plusieurs objets BRep, la structure de données prend aussi en compte plusieurs objets. Ceci explique les différences observées entre les deux tests, notamment pour les temps moyens. Le temps moyen de simulation par rayon décroît lorsque le nombre de subdivisions augmente. Cependant, on observe que la pente de la courbe représentant ce paramètre s'atténue fortement. Ce qui signifie que la méthode est efficace pour un nombre de subdivisions qui n'est pas trop grand. En effet, l'augmentation du nombre de subdivisions tend bien à diminuer le nombre moyen d'intersections rayon-objet, mais, en conséquence, il faut parcourir de plus en plus de régions vides (le pourcentage de voxels vides augmente avec le nombre de subdivisions), ce qui finit par devenir très coûteux. Ce problème est bien connu et largement décrit dans la littérature.

Le tableau ci-dessous rend compte du gain en nombre moyen d'intersections. C'est le nombre moyen d'intersections calculées en moins par rapport au test LU (1,1,1).

système optique	gain en nombre moyen d'intersections					
	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)
(1) alignement de lentilles	1.00	1.84	3.11	7.96	10.24	10.97
(2) aiguille de tableau de bord	1.00	3.13	4.66	7.59	11.97	18.25
(3) compteur d'automobile	1.00	4.27	9.41	18.86	23.45	31.18
(4) ampoule	1.00	4.14	8.49	18.89	26.43	31.18
(5) matrice de lentilles	1.00	4.05	9.71	23.74	56.90	99.67
(6) matrice d'objets	1.00	4.22	12.39	29.40	71.98	133.26
(7) système optique	1.00	2.92	6.65	12.07	47.80	72.18
(8) système d'éclairage d'intérieur	1.00	5.11	9.39	23.14	38.50	67.01

Traçons la courbe associée à ce paramètre, en indiquant pour chaque système optique le nombre d'objets considérés pour la localisation.



Pour affiner ces observations, il faut regarder de plus près les paramètres rendant compte des complexités en temps et en mémoire. Les deux tableaux ci-après récapitulent les valeurs des deux paramètres suivants :

- le temps moyen de simulation par rayon,
- la taille de la structure de données.

L'intérêt d'observer conjointement ces deux paramètres réside dans le fait que, pour rendre utilisable la méthode proposée, l'on doit rechercher la subdivision uniforme qui procure la plus petite complexité en temps, tout en restant acceptable pour la complexité en mémoire.

système optique	temps moyen de simulation par rayon (ms)					
	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)
(1) alignement de lentilles	1.94	1.19	0.78	0.47	0.44	0.47
(2) aiguille de tableau de bord	19.47	18.18	16.80	16.43	15.42	14.75
(3) compteur d'automobile	16.67	15.19	14.74	9.02	8.76	8.55
(4) ampoule	32.46	27.92	24.96	22.20	21.73	21.54
(5) matrice de lentilles	9.57	2.85	1.53	0.98	0.75	0.76
(6) matrice d'objets	45.68	13.28	6.91	4.79	3.86	3.61
(7) système optique	102.50	36.09	17.48	10.81	4.65	4.09
(8) système d'éclairage d'intérieur	117.81	55.58	48.28	42.44	40.37	38.94

système optique	taille de la structure de données (octets)					
	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)
(1) alignement de lentilles	240	1 560	7 624	32 800	108 996	372 216
(2) aiguille de tableau de bord	296	1 572	4 312	16 452	77 944	325 536
(3) compteur d'automobile	316	1 608	8 684	45 940	213 052	1 234 356
(4) ampoule	528	2 368	9 808	46 352	310 912	2 202 816
(5) matrice de lentilles	1 308	3 420	6 672	23 152	88 232	518 436
(6) matrice d'objets	2 508	6 780	16 152	63 548	301 484	1 825 572
(7) système optique	4 224	14 456	31 056	110 064	426 432	2 548 480
(8) système d'éclairage d'intérieur	8 456	19 496	28 616	87 288	420 360	2 891 608

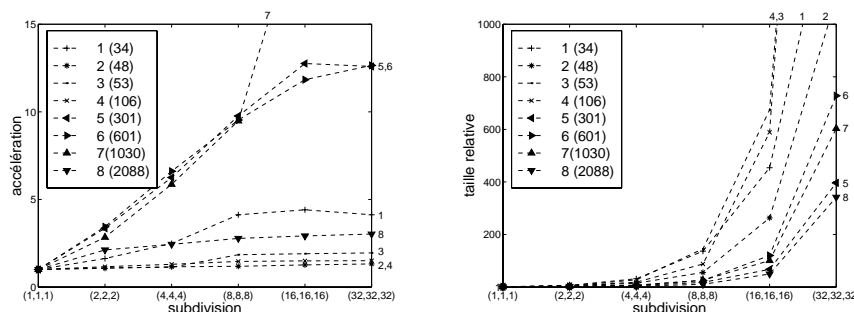
Pour analyser ces deux paramètres, calculons les valeurs des deux paramètres relatifs au test LU (1,1,1) pour les différents systèmes optiques considérés :

- l'accélération, qui est le temps moyen relatif de simulation par rayon,
- la taille relative de la structure de données.

système optique	accélération (réf. LU (1,1,1))					
	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)
(1) alignement de lentilles	1.00	1.63	2.49	4.13	4.41	4.13
(2) aiguille de tableau de bord	1.00	1.07	1.16	1.19	1.26	1.32
(3) compteur d'automobile	1.00	1.10	1.13	1.85	1.90	1.95
(4) ampoule	1.00	1.16	1.30	1.46	1.49	1.51
(5) matrice de lentilles	1.00	3.36	6.25	9.77	12.76	12.59
(6) matrice d'objets	1.00	3.44	6.61	9.54	11.83	12.65
(7) système optique	1.00	2.84	5.86	9.48	22.04	25.06
(8) système d'éclairage d'intérieur	1.00	2.12	2.44	2.78	2.92	3.03

système optique	taille relative de la structure de données (réf. LU (1,1,1))					
	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)
(1) alignement de lentilles	1.00	6.50	31.80	136.70	454.20	1 550.90
(2) aiguille de tableau de bord	1.00	5.30	14.60	55.60	263.30	1 099.80
(3) compteur d'automobile	1.00	5.10	27.50	145.40	674.20	3 906.20
(4) ampoule	1.00	4.50	18.60	87.80	588.80	4 172.00
(5) matrice de lentilles	1.00	2.60	5.10	17.70	67.50	396.40
(6) matrice d'objets	1.00	2.70	6.40	25.30	120.20	727.90
(7) système optique	1.00	3.40	7.40	26.10	101.00	603.30
(8) système d'éclairage d'intérieur	1.00	2.30	3.40	10.30	49.70	342.00

Traçons les courbes correspondantes en indiquant pour chaque système optique le nombre d'objets considérés pour la localisation.



Les courbes de l'accélération ne sont pas, comme on aurait pu naïvement le penser, empilées dans le sens croissant des nombres d'objets. Les systèmes optiques 2 à 4 (de 48 à 106 objets) ont des

courbes similaires. Ce n'est pas étonnant puisque les nombres d'objets sont semblables ainsi que les répartitions géométriques caractérisées par les courbes de répartition de la dimension et de l'aire de la surface optique. De même, les systèmes optiques 5 et 6 (301 et 601 objets) ont des courbes similaires, et ce pour les mêmes raisons. La courbe associée au système optique 1 (34 objets) présente un pallier à partir du test LU (8,8,8). Celle associée au système optique 7 (1 030 objets) montre une accélération forte à partir du test LU (16,16,16). Enfin, l'accélération pour le système optique 8 (2 088 objets) est quasiment constante à partir du test LU (2,2,2).

On ne peut donc pas mettre en évidence une relation uniquement entre le nombre d'objets d'un système optique et le nombre de subdivisions. L'écart entre les courbes des systèmes optiques 7 et 8 le montre. Bien évidemment, ce qui doit être pris aussi en compte c'est la répartition géométrique des objets, leurs dimensions et leur nature. Le système optique 7 est composé en grande partie d'objets simples (99.7% de plans) dont la dimension relative est grande (près de 90% d'objets dont la dimension relative est comprise entre 0.6 et 0.8). Le système optique 8 est composé d'objets complexes (99.4% de B-splines) dont la dimension relative est petite (près de 60% d'objets dont la dimension relative est comprise entre 0 et 0.1). Pour ce système optique, l'efficacité de la localisation par subdivision uniforme ne peut se faire ressentir qu'à partir d'un grand nombre de subdivisions, car de petits objets complexes engendrent des coûts de calculs d'intersection élevés. Or, lorsque le nombre de subdivisions devient grand, cette méthode de localisation perd en efficacité.

Les courbes de la taille relative montrent, quel que soit le système optique considéré, que la complexité en mémoire relative (par rapport au test LU (1,1,1)) reste acceptable jusqu'au test LU (8,8,8) pour les systèmes optiques 1 à 4 (34 à 106 objets), et jusqu'au test LU (16,16,16) pour les systèmes optiques 5 à 8 (301 à 2 088 objets). Bien que les régions vides, c'est-à-dire n'ayant d'intersection avec aucun objet (au sens de la localisation), ne soient pas enregistrées dans la structure de données, il en résulte une complexité en mémoire qui devient très grande.

Là aussi les courbes ne sont pas empilées dans le sens croissant des nombres d'objets. Cependant, l'influence du nombre d'objets est plus forte que pour les courbes de l'accélération. En effet, on distingue bien deux groupes de systèmes optiques 1 à 4 et 5 à 8. La répartition géométrique des objets doit être prise en compte.

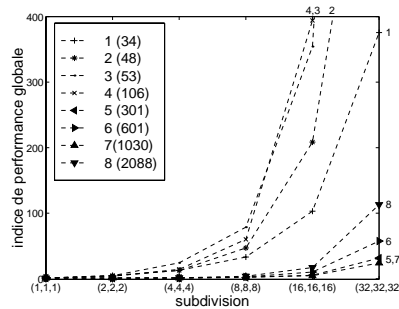
Pour renforcer ces premières analyses, il est intéressant d'observer le rapport entre la taille relative et l'accélération pour chaque système optique et pour chaque subdivision. On introduit un nouveau paramètre pour quantifier cette grandeur, que l'on appellera indice de performance globale, et que l'on notera ρ :

$$\rho = \frac{M_r}{\overline{T}_r} ,$$

où M_r est la complexité en mémoire relative (taille relative de la structure de données) et \overline{T}_r la complexité moyenne en temps relative (accélération). Calculons les différentes valeurs de ρ .

système optique	indice de performance globale					
	LU (1,1,1)	LU (2,2,2)	LU (4,4,4)	LU (8,8,8)	LU (16,16,16)	LU (32,32,32)
(1) alignement de lentilles	1.0	4.0	12.8	33.1	103.0	375.7
(2) aiguille de tableau de bord	1.0	5.0	12.6	46.9	208.5	833.2
(3) compteur d'automobile	1.0	4.6	24.3	78.7	354.3	2 003.5
(4) ampoule	1.0	3.9	14.3	60.0	394.2	2 768.5
(5) matrice de lentilles	1.0	0.8	0.8	1.8	5.3	31.5
(6) matrice d'objets	1.0	0.8	1.0	2.7	10.2	57.5
(7) système optique	1.0	1.2	1.3	2.7	4.6	24.1
(8) système d'éclairage d'intérieur	1.0	1.1	1.4	3.7	17.0	113.0

Traçons les courbes associées à ce nouveau paramètre, en indiquant pour chaque système optique le nombre d'objets considérés pour la localisation.



Les courbes montrent que l'on peut différencier les systèmes optiques 1 à 4 des systèmes optiques 5 à 8. En effet, les systèmes optiques 1 à 4 sont associés à un indice de performance globale acceptable (inférieur à 100) jusqu'au test LU (8,8,8), alors que pour les systèmes optiques 5 à 8 il est acceptable quasiment jusqu'au test LU (32,32,32). C'est logique de trouver une analyse semblable à celle issue des courbes de la taille relative puisque l'influence sur ρ de la complexité en mémoire relative M_r est supérieure à celle de la complexité moyenne en temps relative \bar{T}_r .

L'intérêt de l'indice de performance globale est de donner une valeur limite au delà de laquelle la méthode de localisation par subdivision uniforme n'est plus efficace. En effet, dès que la pente de la courbe de ρ devient grande, à la vue des courbes expérimentales, on ne peut espérer qu'une dégradation rapide de l'efficacité de la méthode. Il reste à fixer un seuil pour ρ . Les expériences menées ici nous amènent à considérer la valeur 100. Bien entendu, ce seuil décrit une limite supérieure relative au niveau des complexités. Il faut en plus veiller à ce que la complexité en mémoire ne soit pas trop grande, au regard des contraintes matérielles.

Hormis le constat classique consistant à dire que le parcours des régions vides devient coûteux lorsque leur nombre croît, ces expériences sur une localisation par subdivision uniforme met en évidence l'intérêt qu'il faut porter à la répartition géométrique des objets, leurs dimensions, mais aussi à leur nature. Lorsque les objets considérés sont complexes, les algorithmes de calcul d'intersection rayon-objet possèdent des critères d'exclusion géométriques, qui se traduisent par le fait que plus un rayon est proche de l'objet, plus le temps de recherche de l'intersection est grand. Or la localisation par subdivision uniforme rapproche les rayons des objets (les calculs d'intersection sont effectués pour des rayons plus proches), et par conséquent accroît le temps moyen d'intersection rayon-objet, ce qui entraîne une perte en efficacité de la méthode lorsque le nombre moyen d'intersections ne diminue pas suffisamment.

Hierarchie de subdivisions uniformes

Une localisation par une subdivision uniforme est peu efficace et très vite limitée par le nombre de subdivisions que l'on peut considérer. Cette limite, comme on vient de le voir, est essentiellement due à deux raisons. L'augmentation du nombre de subdivisions entraîne une augmentation du nombre de régions à parcourir qui devient très vite grand (c'est caractérisé par le nombre moyen de voxels traversés). La taille de la structure de données associée croît aussi très vite. L'intérêt d'utiliser une localisation par hiérarchie de subdivisions uniformes est donc de diminuer au mieux les espaces vides et d'obtenir une structure de données dont la taille reste acceptable.

Pour analyser l'efficacité des tests LHU, il est nécessaire d'observer, d'une part, le temps de pré-calcul (conversion vers le modèle BRep et construction de la structure de données) et, d'autre part, l'indice de performance globale qui est le rapport entre la taille relative et l'accélération. Le tableau suivant récapitule ces différentes valeurs. La taille relative et l'indice de performance globale sont comparés au test LU (1,1,1).

système optique	pré-calcul (s)	accélération		taille relative	performance globale
		réf. SL	réf. LU (1,1,1)		
(1) alignement de lentilles	2	0.96	3.18	23.25	7.3
(2) aiguille de tableau de bord	3	1.15	1.18	70.58	59.8
(3) compteur d'automobile	1	1.76	1.81	109.37	60.4
(4) ampoule	2	1.29	1.43	158.15	110.6
(5) matrice de lentilles	157	3.04	10.29	261.02	25.4
(6) matrice d'objets	26	10.77	12.00	354.08	29.5
(7) système optique	37	19.77	25.85	8 114.75	313.9
(8) système d'éclairage d'intérieur	31	2.45	3.15	1 647.72	523.1

Les coûts en temps du pré-calcul, comprenant la conversion de la scène géométrique vers le modèle BRep et la construction de la structure de données, sont faibles. Le plus grand étant pour le système optique 5 (101 objets à convertir) où l'on atteint plus de 1 minute et 30 secondes. Comparés au coût en temps des simulations, qui peuvent aller jusqu'à plusieurs heures pour un résultat acceptable au niveau de la précision numérique sur les cartes d'observation, ce temps est négligeable.

La lecture des paramètres observés (accélération, taille relative et indice de performance globale) montre que l'efficacité d'une hiérarchie de subdivisions uniformes dépend du système optique considéré. Pour le système optique 1 (34 objets), le test LHU est plus efficace que les tests LU, notamment au regard de l'indice de performance globale. Pour les systèmes optiques 2 et 3 (48 et 53 objets), les tests LHU et LU ont des performances similaires jusqu'au test LU (8,8,8). Ensuite, l'accroissement du nombre de régions vides entraîne une forte augmentation de la taille de la structure de données pour les test LU, et par conséquent un indice de performance globale trop élevé. Pour le système optique 4 (106 objets), la configuration LU (8,8,8) est plus efficace que la configuration LHU, pour laquelle la taille de la structure de données associée est trop importante. Pour les systèmes optiques 5 et 6 (301 et 601 objets), les performances des deux méthodes sont comparables. Pour les systèmes optiques 7 et 8 (1 030 et 2 088 objets), l'indice de performance globale du test LHU est beaucoup plus élevé que ceux des test LU. Là encore, la taille de la structure de données associée à la hiérarchie de subdivisions uniformes est trop importante.

Ces tests montrent que l'efficacité d'une hiérarchie de subdivisions uniformes n'est pas uniquement liée au nombre d'objets. Pour les systèmes optiques testées, il apparaît clairement que la seuil choisi pour la profondeur (5 niveaux) est trop grand. D'autres tests, non présentés ici, ont montrés que l'on pourrait se limiter à deux, voire trois, niveaux pour la profondeur. Néanmoins, dans tous les cas les accélérations obtenues avec une localisation par hiérarchie de subdivisions uniformes sont très intéressantes par rapport à celles issues d'une localisation par subdivisions uniformes.

3.4 Conclusion

Ce chapitre présente la solution envisagée dans le contexte industriel pour accélérer les calculs d'intersection. La prise en compte d'un cahier des charges contraignant a fortement orienté le choix vers une méthode de localisation spatiale par subdivision uniforme. Néanmoins, les résultats obtenus sont très satisfaisants. Leur analyse met en évidence la difficulté de choisir les bons paramètres pour une subdivision uniforme et encore plus pour une hiérarchie de subdivisions uniformes. Les différentes études connues à ce jour ne fournissent que des solutions intuitives ou empiriques. En effet, l'évaluation des complexités en mémoire et en temps pour ces méthodes est un problème ouvert.

Dans la suite, on se placera dans un contexte plus théorique, et on proposera une méthode pour évaluer a priori ces complexités et ainsi permettre un choix systématique des paramètres de localisation.

Chapitre 4

Évaluation des complexités dans une subdivision uniforme

La caractérisation des complexités en mémoire et en temps d'un algorithme de recherche d'intersections par une méthode de localisation des objets est un problème ouvert. La plupart des solutions proposées sont empiriques. En effet, les auteurs utilisent des observations faites sur des scènes types pour déterminer des paramètres de localisation. Plus particulièrement, en ce qui concerne l'algorithme étudié au chapitre précédent (algorithme 12 page 116), c'est-à-dire la recherche d'intersections par localisation des objets dans une subdivision uniforme, le même type d'approche est proposé, et les travaux visant une approche formelle du problème sont quasiment inexistantes. Néanmoins, Cleary et Wyvill [18] se sont lancés dans une voie très intéressante. Ils ont été les premiers à proposer une expression des complexités en fonction de certaines caractéristiques géométriques des objets et des paramètres de localisation.

La première difficulté pour l'évaluation des complexités est de poser correctement le problème. Les expressions de ces complexités font intervenir des paramètres propres à la scène géométrique, aux propriétés optiques des objets et à la subdivision uniforme considérées. La seconde difficulté est de donner une expression exacte voire une estimation de ces paramètres. C'est ce que nous proposons de faire dans ce chapitre, en donnant une caractérisation des complexités qui prend en compte la géométrie de la scène ainsi que la propriété optique d'opacité et les paramètres de la subdivision uniforme considérée.

Le premier paragraphe présente les stratégies classiques pour définir a priori les paramètres d'une hiérarchie de subdivisions uniformes. Le second paragraphe est consacré à l'évaluation du nombre de régions d'une subdivision uniforme traversées par un objet. Les deux paragraphes suivants proposent des expressions, d'une part, pour la complexité en mémoire des structures de données présentées aux chapitres 2 et 3 et, d'autre part, pour la complexité moyenne en temps relative aux algorithmes du chapitre 3. Enfin, les deux derniers paragraphes donnent un moyen de choisir a priori les paramètres d'une subdivision uniforme ainsi que ceux d'une hiérarchie de subdivisions uniformes.

4.1 Choix classiques des paramètres de subdivision

Les méthodes basées sur les subdivisions uniformes sont caractérisées à partir de deux paramètres essentiels :

- le nombre de subdivisions dans chaque direction,
- la profondeur de la hiérarchie.

Leur objectif est d'atteindre une complexité en temps la plus faible possible, tout en contrôlant la complexité en mémoire. Le choix de ces paramètres est un problème difficile, qui n'a pas encore trouvé de solution optimale. En effet, pour choisir les bons paramètres il faudrait prendre en compte les

caractéristiques géométriques des objets et les propriétés optiques associées. C'est utopique au regard de la quantité d'information à considérer pour un système optique quelconque. De plus, la corrélation entre les paramètres de chaque subdivision uniforme et la profondeur de la hiérarchie rend très difficile ce choix. En effet, ce n'est pas en prenant les paramètres qui minimisent les complexités en temps pour chaque subdivision uniforme de la hiérarchie que l'on minimise nécessairement la complexité en temps globale.

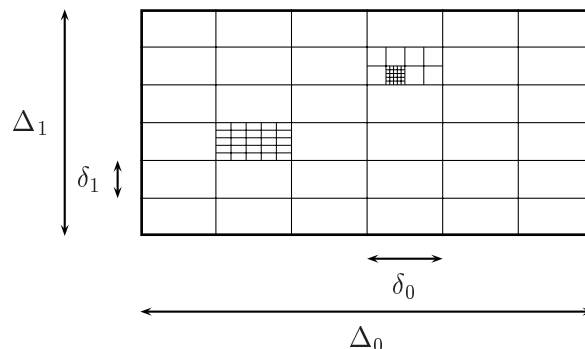


Fig. 4.1 – Hiérarchie de subdivisions uniformes.

Faisons un tour d'horizon des stratégies existantes. Dans la plupart des travaux, le nombre de subdivisions dans chaque direction est déterminé par l'expression suivante :

$$n_i = \frac{\Delta_i}{\delta_i} = \lceil \alpha_i N^{1/3} \rceil ,$$

où Δ_i est la longueur dans la direction i ($i = 0, 1, 2$) du parallélépipède à subdiviser, α_i est un coefficient dépendant de la méthode, N le nombre d'objets et $\lceil \cdot \rceil$ l'opérateur numérique d'arrondi (valant 1 au minimum pour être cohérent avec un nombre de subdivisions). Le choix le plus classique est $\alpha_i = 1$ dans toutes les directions. Cette stratégie ne prend en compte que le nombre d'objets. C'est optimal si leur répartition est uniforme et si leurs dimensions sont similaires. Cazals, Drettakis et Puech [15] proposent de prendre $\alpha_i = \sqrt{2}$ lorsque le nombre d'objets N est assez grand. C'est un choix issu d'observations expérimentales. Klimaszewski et Sederberg [59] prennent en compte la taille du parallélépipède à subdiviser et posent :

$$\alpha_i = \left(\frac{\Delta_i^2}{\Delta_{i+1} \Delta_{i+2}} \right)^{1/3} ,$$

où Δ_{i+1} et Δ_{i+2} doivent être compris comme $\Delta_{(i+1) \bmod 3}$ et $\Delta_{(i+2) \bmod 3}$.

Dans un registre différent, Scherson et Caspary [74] proposent une approche formelle qui consiste à exprimer la complexité en temps de recherche de la première intersection entre un rayon et un objet (polygone plan) en fonction, entre autres, des paramètres de localisation. La minimisation de l'expression de la complexité donne alors le nombre optimal, en un certain sens, de subdivisions dans chaque direction :

$$\alpha_i = \left(\frac{2T_O}{T_V} \right)^{1/3} ,$$

où T_O est la complexité en temps de recherche de la première intersection entre un rayon et un objet et T_V est la complexité en temps de parcours d'une région de la subdivision uniforme considérée. Cette approche est limitée à des subdivisions uniformes cubiques et ne prend pas en compte les caractéristiques géométriques des objets. Dans une voie similaire, Cleary et Wyvill [18] expriment les complexités en temps et en mémoire pour des objets constitués de polygones plans. Leur démarche, bien que novatrice,

ne prend en compte que des subdivisions uniformes cubiques et la géométrie des objets est considérée par rapport à une moyenne sur leur orientation. Khachan [57] donne une méthode permettant de choisir des critères de subdivision des objets (des surfaces fermées au sens où elles définissent un objet volumique) de manière à générer des variétés discrètes qui sont caractérisées à l'aide de critères topologiques locaux (au sens de la topologie discrète). Cette méthode permet, entre autres, de séparer au mieux les objets, dans le sens où chaque composante connexe définit une variété. Cependant, elle a été développée pour le cas implicite, elle nécessite une mise en œuvre complexe et les structures de données sont très gourmandes en mémoire (dans le cas uniforme).

La profondeur d'une hiérarchie de subdivisions uniformes est le plus souvent connue à posteriori, elle dépend du nombre d'objets rencontrés au fur et à mesure que l'on subdivise. Un seuil est fixé pour contrôler en partie la complexité en mémoire. Une région n'est alors subdivisée que si elle contient un nombre d'objets supérieur à ce seuil. Par conséquent, les caractéristiques géométriques des objets ainsi que leurs positions relatives ne sont pas prises en compte pour la mise en place de la hiérarchie (indépendamment des paramètres de chaque subdivision uniforme), autrement dit chaque nœud de l'arbre des subdivisions uniformes est traité indépendamment des autres nœuds. MacDonald et Booth [63] poursuivent dans la voie empruntée par Goldsmith et Salmon [45] et proposent de construire un arbre de subdivisions uniformes en tentant de contrôler la répartition des complexités en temps pour chaque nœud de l'arbre. Pour ce faire, ils considèrent une distribution uniforme des rayons dans $\mathbb{R}^3 \times \mathcal{S}^2$ et expriment la complexité en temps sur l'arbre en fonction des aires des parallélépipèdes à partir desquels sont définis les différentes subdivisions uniformes et des nombres d'objets liés à chacun de ces parallélépipèdes. Cependant, leur méthode ne prend pas en compte les caractéristiques géométriques des objets.

4.2 Nombre de régions traversées par un objet

4.2.1 Caractérisation

Soit O un ensemble de points de \mathbb{R}^3 décrivant un objet, au sens des objets définis au chapitre 2. Soit R une région parallélépipédique contenant O . Soit $\mathcal{S} = \{R_k\}_{k=1}^n$ une subdivision uniforme de R (paragraphe 3.2.1), où les R_k sont des parallélépipèdes. On se pose alors le problème suivant : évaluer le nombre de régions de \mathcal{S} traversées par O , noté m_O et défini par :

$$m_O = \text{card}(L_O) ,$$

où, de manière analogue à la définition 6 page 96, L_O est la liste des indices des régions de \mathcal{S} traversées par O :

$$L_O = \{k \in \{1, \dots, n\} \text{ tel que } R_k \cap O \neq \emptyset\} .$$

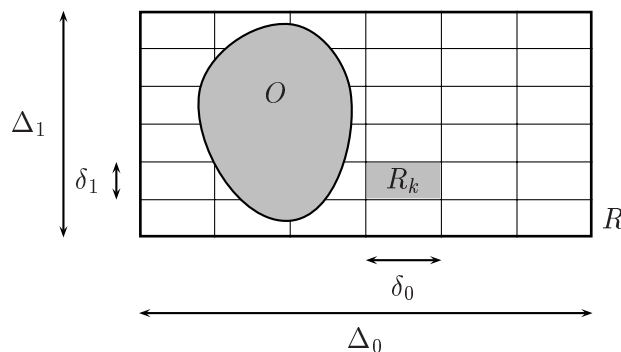


Fig. 4.2 – Notations pour l'évaluation du nombre de régions traversées par un objet dans une subdivision uniforme.

L'évaluation de m_O est un problème non trivial. La géométrie discrète ne fournit que quelques résultats pour des objets simples (segment, triangle, cercle en dimension 2, hyperplan en dimension quelconque). De plus, il est facile de remarquer que le nombre recherché n'est pas invariant si l'on applique à l'objet O une translation. Soit G est un opérateur géométrique de translation. Le nombre de régions de \mathcal{S} traversées par l'objet GO est par définition :

$$L_{GO} = \{k \in \{1, \dots, n\} \text{ tel que } R_k \cap (GO) \neq \emptyset\} .$$

On a pas nécessairement $m_{GO} = m_O$. C'est encore plus remarquable si l'on considère un opérateur géométrique plus général (rotation, homothétie). C'est pourquoi nous proposons une approche visant à donner une estimation en moyenne de la quantité m_O , basée sur la périodicité de la fonction m_{GO} pour des opérateurs géométriques de translation. La proposition suivante caractérise l'invariance du nombre de régions traversées pour une certaine classe de translations.

Proposition 15

Soit $\mathcal{S} = \{R_k\}_{k \in \mathbb{Z}}$ une subdivision uniforme de \mathbb{R}^3 . Soit G un opérateur géométrique de translation caractérisé par un vecteur $(v_0, v_1, v_2) \in \mathbb{R}^3$ tel que :

$$v_i = \alpha_i \delta_i ,$$

où $\alpha_i \in \mathbb{Z}$ et δ_i est la dimension d'une région dans la direction i . Alors, le nombre de régions de \mathcal{S} traversées par l'objet GO est identique au nombre de régions de \mathcal{S} traversées par l'objet O :

$$m_{GO} = m_O .$$

□

Preuve C'est évident, il suffit de constater que la translation de O revient à déplacer la subdivision uniforme. Le choix de G implique que les deux subdivisions uniformes coïncident, et, par conséquent, que $m_{GO} = m_O$. ■

La proposition précédente est très intéressante en ce sens qu'elle caractérise la périodicité de la fonction m_{GO} . Soit $\mathcal{G}_{\mathcal{S}, \mathbb{Z}}$ l'ensemble des opérateurs géométriques de translation suivant :

$$\mathcal{G}_{\mathcal{S}, \mathbb{Z}} = \{G \text{ translation de vecteur } (\alpha_0 \delta_0, \alpha_1 \delta_1, \alpha_2 \delta_2) \text{ tel que } \alpha_i \in \mathbb{Z}\} .$$

La proposition 15 signifie que :

$$\forall G \in \mathcal{G}_{\mathcal{S}, \mathbb{Z}} , m_{GO} = m_O .$$

En d'autres termes, la fonction m_{GO} est entièrement caractérisée pour l'ensemble de translations $\mathcal{G}_{\mathcal{S}, 0}$ défini par :

$$\mathcal{G}_{\mathcal{S}, 0} = \{G \text{ translation de vecteur } (\alpha_0 \delta_0, \alpha_1 \delta_1, \alpha_2 \delta_2) \text{ tel que } \alpha_i \in [-1/2, 1/2]\} .$$

Définir une moyenne de la fonction m_{GO} pour des translations $G \in \mathcal{G}_{\mathcal{S}, 0}$ revient à sommer toutes les valeurs de la fonction m_{GO} pour tous les G considérés puis à diviser par la mesure de l'ensemble $\mathcal{G}_{\mathcal{S}, 0}$. Cependant, on ne sait pas calculer m_{GO} pour un G donné, et, de plus, la notion de mesure sur l'ensemble $\mathcal{G}_{\mathcal{S}, 0}$ est à préciser. Néanmoins, on peut s'en sortir si l'on remarque que l'union des objets GO pour des translations de $\mathcal{G}_{\mathcal{S}, 0}$ s'écrit de la manière suivante :

$$\bigcup_{G \in \mathcal{G}_{\mathcal{S}, 0}} (GO) = R_0 + O ,$$

où R_0 est le parallélépipède défini par :

$$R_0 = \prod_{i=0}^2 \left[-\frac{\delta_i}{2}, \frac{\delta_i}{2} \right],$$

et $R_0 + O$ est l'ensemble des points de \mathbb{R}^3 caractérisé par (figure 4.3) :

$$R_0 + O = \{x \in \mathbb{R}^3 \text{ tel que } x = y + z \text{ avec } y \in R_0 \text{ et } z \in O\}.$$

$R_0 + O$ est appelé somme de Minkowski des ensembles R_0 et O .

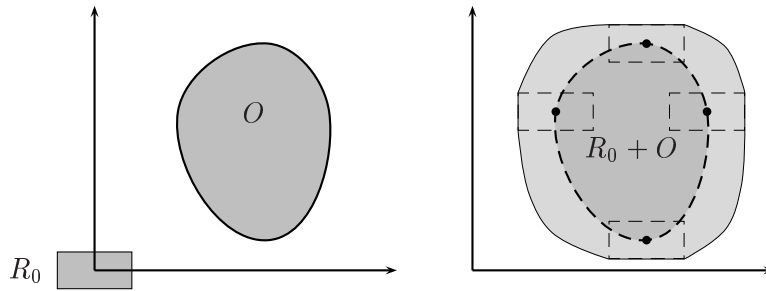


Fig. 4.3 – Somme de Minkowski d'une région R_0 et d'un objet O .

La définition suivante caractérise le nombre moyen de régions d'une subdivision uniforme traversées par un objet.

Définition 7

Le nombre moyen de régions d'une subdivision uniforme $\mathcal{S} = \{R_k\}_{k \in \mathbb{Z}}$ traversées par un objet O est donné par le nombre suivant :

$$\overline{m}_O = \frac{\text{vol}(R_0 + O)}{\text{vol}(R_0)}, \quad (4.1)$$

où R_0 est le parallélépipède défini par :

$$R_0 = \prod_{i=0}^2 \left[-\frac{\delta_i}{2}, \frac{\delta_i}{2} \right]. \quad (4.2)$$

□

Remarque

La définition précédente s'appuie sur une subdivision uniforme de \mathbb{R}^3 . Si l'on considère une subdivision uniforme d'un parallélépipède R , alors la définition reste valable pour peu que l'union des objets GO soit incluse dans R . Cependant, dans le cas contraire, cela ne pose pas vraiment de problème dans le sens où \overline{m}_O est une valeur moyenne.

□

Les objets qui nous intéressent sont de deux types :

- les segments, qui sont issus des rayons considérés,
- les objets surfaciques ou volumiques, dont la surface optique associée (définition 5 page 96) est une surface simple continue et dérivable par morceaux.

Pour le segment (paragraphe 4.2.2), une écriture explicite est envisageable, et, par simple extension, une majoration peut être donnée pour une ligne brisée (succession de segments) et donc pour un objet linéique. Pour les autres objets c'est beaucoup plus compliqué. Pour la localisation spatiale, on s'intéresse aux intersections entre un objet $\Upsilon(O)$ (définition 6 page 96) et les régions de \mathcal{S} , où $\Upsilon(O)$ est soit la surface optique associée à l'objet O , soit un objet englobant O , le plus souvent une boîte englobante $B(O)$. Pour une surface optique (paragraphe 4.2.3), l'évaluation ne peut se faire explicitement dans le cas général, on considère alors une approximation de la surface par des polygones plans, et l'on donne une majoration du nombre de régions traversées. Pour une boîte englobante (paragraphe 4.2.4), une écriture explicite est donnée. Dans tous les cas, le résultat recherché est approché par une expression du nombre de régions de \mathcal{S} traversées en fonction des caractéristiques géométriques de l'objet géométrique O considéré. Pour ne pas prendre en compte les problèmes de bord induit par le parallélogramme R , on se contentera d'une subdivision uniforme de \mathbb{R}^3 .

4.2.2 Objet linéique

Soit $\mathcal{S} = \{R_k\}_{k \in \mathbb{Z}}$ une subdivision uniforme de \mathbb{R}^3 . Soit Γ un objet linéique. Notons m_Γ le nombre de régions de \mathcal{S} traversées par Γ (figure 4.4).

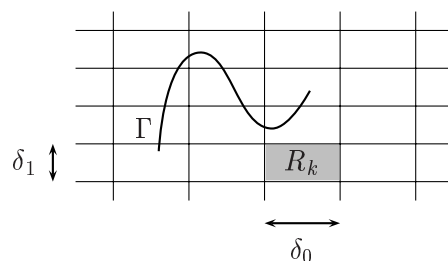


Fig. 4.4 – Notations pour l'évaluation du nombre de régions traversées par un objet linéique dans une subdivision uniforme.

L'objectif de ce paragraphe est d'évaluer le nombre m_Γ . L'approche par une estimation moyenne du nombre de régions traversées par un segment, au sens de la définition 7, est une généralisation du résultat donné par Cleary et Wyvill [18]. Pour calculer le nombre moyen de régions traversées par un segment les auteurs utilisent ce qu'ils appellent le volume augmenté d'une région de \mathcal{S} et du segment considéré, qui est en fait la somme de Minkowski de ces deux éléments. Le rapport des volumes de la somme de Minkowski et d'une région de \mathcal{S} donne une estimation du nombre moyen de régions traversées. Cependant, leur résultat n'est donné que pour des régions cubiques, avec des nombres de subdivisions identiques dans toutes les directions et il est exprimé par rapport à une moyenne sur l'orientation du segment.

On donnera tout d'abord le résultat pour un segment, puis on s'intéressera à une ligne brisée.

Segment

Soit $A = [x, x + \ell w]$ un segment, et notons m_A le nombre de régions de \mathcal{S} traversées par ce segment (figure 4.5).

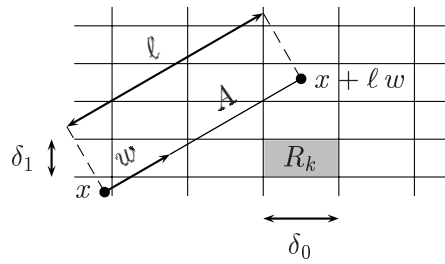


Fig. 4.5 – Notations pour l'évaluation du nombre de régions traversées par un segment dans une subdivision uniforme.

La proposition suivante donne une estimation du nombre moyen de régions de \mathcal{S} traversées par un segment A .

Proposition 16

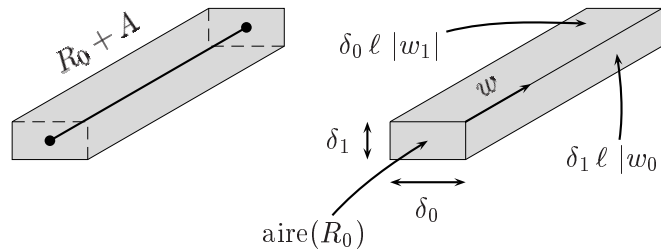
Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^3 traversées par un segment A de longueur ℓ et d'orientation w données est donné par le nombre suivant :

$$\overline{m}_A = 1 + \ell \sum_{i=0}^2 \frac{|w_i|}{\delta_i}, \quad (4.3)$$

où w_i est la coordonnée dans la direction i du vecteur w .

□

Preuve Partant de la définition 7, il suffit d'écrire le volume de la somme de Minkowski $R_0 + A$. Pour ce faire, on le décompose en quatre parties : le volume de R_0 et le volume de trois parallélépipèdes. Par analogie, l'aire de la somme de Minkowski d'un rectangle R_0 et d'un segment A situés dans \mathbb{R}^2 se décompose en trois parties : l'aire de R_0 et l'aire de deux parallélogrammes.



Ainsi, le volume résultant s'écrit :

$$\text{vol}(R_0 + A) = \text{vol}(R_0) + \sum_{i=0}^2 \delta_{i+1} \delta_{i+2} \ell |w_i|,$$

où δ_{i+1} et δ_{i+2} doivent être compris comme $\delta_{(i+1) \bmod 3}$ et $\delta_{(i+2) \bmod 3}$. En divisant par le volume de R_0 , on trouve l'expression proposée pour \overline{m}_A .

■

Pour montrer que l'expression 4.3 est une généralisation du résultat donné par Cleary et Wyvill [18], commençons par supposer que la taille des régions de la subdivision uniforme considérée est identique dans toutes les directions :

$$\delta_i = \delta.$$

Ainsi 4.3 s'écrit de la manière suivante :

$$\bar{m}_A = 1 + \frac{\ell}{\delta} \sum_{i=0}^2 |w_i| .$$

La moyenne par rapport à l'ensemble des directions de la sphère unité de ce nombre s'écrit :

$$\langle \bar{m}_A \rangle = \frac{1}{4\pi} \int_{\mathcal{S}^2} \bar{m}_A d\sigma(w) .$$

On remarque que :

$$\int_{\mathcal{S}^2} \sum_{i=0}^2 |w_i| d\sigma(w) = 6\pi .$$

Finalement, le nombre moyen de régions d'une subdivision uniforme traversées par un segment de longueur ℓ s'écrit :

$$\langle \bar{m}_A \rangle = 1 + \frac{3}{2\delta} \ell .$$

On retrouve le résultat donné dans [18]. L'expression 4.3 est donc bien une généralisation de ce résultat.

Ligne brisée

Soit Γ une ligne brisée simple, constituée d'un ensemble fini de segments (figure 4.6) :

$$\Gamma = \bigcup_{i=1}^{n_{\text{seg}}} A_i ,$$

où les segments A_i sont définis comme suit :

$$\begin{cases} A_i = [x^i, x^{i+1}] \\ x^{i+1} = x^i + \ell_i w^i \end{cases} .$$

Si la ligne brisée est fermée alors $x^1 = x^{n_{\text{seg}}}$.

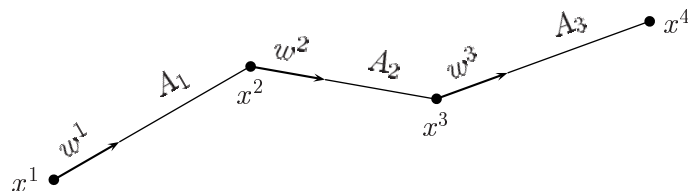


Fig. 4.6 – Notations pour une ligne brisée composée de segments dans \mathbb{R}^3 .

Soit m_Γ le nombre de régions de \mathcal{S} traversées par la ligne brisée Γ . De manière analogue au cas du segment, et pour les mêmes raisons, on cherchera plutôt à évaluer le nombre moyen \bar{m}_Γ de régions de \mathcal{S} traversées. Pour ce faire, on commence par donner un résultat sur la mesure du volume d'une union de sous-ensembles de \mathbb{R}^3 .

Lemme 3

Soient E_1, \dots, E_n des sous-ensembles de \mathbb{R}^3 mesurables (au sens de la mesure de Lebesgue). Alors, le volume de leur union s'écrit :

$$\text{vol} \left(\bigcup_{i=1}^n E_i \right) = \sum_{i=1}^n \left(\text{vol}(E_i) - \text{vol} \left(\bigcup_{j>i} (E_i \cap E_j) \right) \right). \quad (4.4)$$

□

Preuve On raisonne par récurrence sur le nombre de sous-ensembles considérés. Commençons par prendre $n = 2$. Soient E_1, E_2 deux sous-ensembles de \mathbb{R}^3 mesurables. On a alors la relation triviale suivante :

$$\begin{aligned} \text{vol}(E_1 \cup E_2) &= \text{vol}(E_1) + \text{vol}(E_2) - \text{vol}(E_1 \cap E_2) \\ &= (\text{vol}(E_1) - \text{vol}(E_1 \cap E_2)) + \text{vol}(E_2). \end{aligned}$$

La relation 4.4 est donc vérifiée pour $n = 2$. Maintenant, supposons cette relation vraie au rang n et prouvons la au rang $n + 1$. Soient E_1, \dots, E_{n+1} des sous-ensembles de \mathbb{R}^3 mesurables. Alors, on écrit le volume de l'union de ces ensembles sous la forme suivante :

$$\text{vol} \left(\bigcup_{i=1}^{n+1} E_i \right) = \text{vol} \left(\bigcup_{i=1}^n E_i \right) + \text{vol}(E_{n+1}) - \text{vol} \left(\left(\bigcup_{i=1}^n E_i \right) \cap E_{n+1} \right).$$

Le premier terme s'écrit en utilisant l'hypothèse de récurrence, c'est-à-dire la relation 4.4 écrite au rang n . Le troisième terme se réécrit en distribuant l'intersection sur l'union. Ainsi on obtient :

$$\begin{aligned} \text{vol} \left(\bigcup_{i=1}^{n+1} E_i \right) &= \sum_{i=1}^n \left(\text{vol}(E_i) - \text{vol} \left(\bigcup_{j>i} (E_i \cap E_j) \right) \right) + \text{vol}(E_{n+1}) - \text{vol} \left(\bigcup_{i=1}^n (E_i \cap E_{n+1}) \right) \\ &= \sum_{i=1}^{n+1} \left(\text{vol}(E_i) - \text{vol} \left(\bigcup_{j>i} (E_i \cap E_j) \right) \right). \end{aligned}$$

La relation 4.4 est donc vérifiée au rang $n + 1$, par conséquent la récurrence s'enclenche. ■

La proposition suivante donne une majoration du nombre moyen de régions de \mathcal{S} traversées par une ligne brisée.

Proposition 17

Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^3 traversées par une ligne brisée simple Γ , constituée d'un ensemble fini de segments $\{A_i\}_{i=1}^{n_{\text{seg}}}$ est majoré de la manière suivante :

$$\overline{m}_\Gamma \leq 1 + \sum_{i=1}^{n_{\text{seg}}} (\overline{m}_{A_i} - 1), \quad (4.5)$$

où \overline{m}_{A_i} est le nombre moyen de régions de \mathcal{S} traversées par le segment A_i . □

Preuve On part de la définition 7, et on calcule le volume de l'ensemble $R_0 + \Gamma$. Pour ce faire, on écrit l'ensemble sous la forme suivante :

$$R_0 + \Gamma = R_0 + \bigcup_{i=1}^{n_{\text{seg}}} A_i = \bigcup_{i=1}^{n_{\text{seg}}} (R_0 + A_i).$$

D'après le lemme 3, le volume recherché s'écrit de la manière suivante :

$$\text{vol}(R_0 + \Gamma) = \sum_{i=1}^{n_{\text{seg}}} \left(\text{vol}(R_0 + A_i) - \text{vol} \left(\bigcup_{j>i} ((R_0 + A_i) \cap (R_0 + A_j)) \right) \right) .$$

Pour tout $i \in \{1, \dots, n_{\text{seg}} - 1\}$, le second terme dans la somme peut être minoré de la manière suivante :

$$\text{vol} \left(\bigcup_{j>i} ((R_0 + A_i) \cap (R_0 + A_j)) \right) \geq \text{vol}((R_0 + A_i) \cap (R_0 + A_{i+1})) .$$

Cette minoration revient à ne pas tenir compte des éventuelles intersections générées par des segments de Γ non consécutifs. Ainsi, le volume recherché est majoré de la manière suivante :

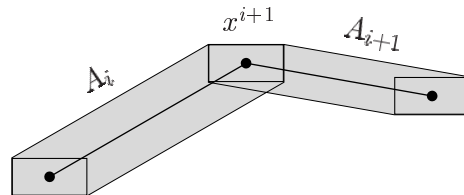
$$\text{vol}(R_0 + \Gamma) \leq \sum_{i=1}^{n_{\text{seg}}} \text{vol}(R_0 + A_i) - \sum_{i=1}^{n_{\text{seg}}-1} \text{vol}((R_0 + A_i) \cap (R_0 + A_{i+1})) .$$

L'intersection entre deux segments consécutifs A_i et A_{i+1} est le point x^{i+1} . On remarque alors que :

$$((R_0 + A_i) \cap (R_0 + A_{i+1})) \supset (R_0 + \{x^{i+1}\}) ,$$

ce qui entraîne :

$$\text{vol}((R_0 + A_i) \cap (R_0 + A_{i+1})) \geq \text{vol}(R_0) .$$



On a finalement la majoration suivante :

$$\text{vol}(R_0 + \Gamma) \leq \sum_{i=1}^{n_{\text{seg}}} \text{vol}(R_0 + A_i) - (n_{\text{seg}} - 1) \text{vol}(R_0) .$$

En divisant par le volume de R_0 on obtient :

$$\overline{m}_\Gamma \leq \sum_{i=1}^{n_{\text{seg}}} \overline{m}_{A_i} - (n_{\text{seg}} - 1) .$$

Le membre de droite peut se réécrire sous la forme :

$$\sum_{i=1}^{n_{\text{seg}}} (\overline{m}_{A_i} - 1) + n_{\text{seg}} - (n_{\text{seg}} - 1) = 1 + \sum_{i=1}^{n_{\text{seg}}} (\overline{m}_{A_i} - 1) .$$

D'où le résultat proposé. ■

4.2.3 Surface optique

Soit $\mathcal{S} = \{R_k\}_{k \in \mathbb{Z}}$ une subdivision uniforme de \mathbb{R}^3 . Soit Σ une surface, et notons m_Σ le nombre de régions de \mathcal{S} traversées par Σ (figure 4.7).

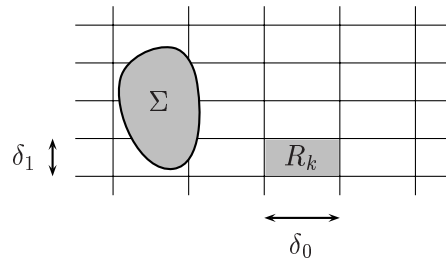


Fig. 4.7 – Notations pour l'évaluation du nombre de régions traversées par une surface dans une subdivision uniforme.

L'objectif de ce paragraphe est d'évaluer le nombre m_Σ . L'approche présentée dans ce paragraphe constitue une généralisation des résultats donnés par Cleary et Wyvill [18]. Les auteurs utilisent la notion de volume augmenté, autrement dit de somme de Minkowski de deux ensembles, de manière analogue à l'évaluation du nombre moyen de régions traversées par un segment. Cependant le nombre qu'ils donnent n'est valable que pour des régions cubiques, avec des nombres de subdivisions identiques dans toutes les directions, et il est exprimé par rapport à une moyenne sur l'orientation de la surface. On raisonnera dans un premier temps dans \mathbb{R}^2 , puis on prolongera les résultats dans \mathbb{R}^3 . Pour chaque dimension on commencera par énoncer deux résultats pour des polygones plans convexes et non convexes. Enfin, on énoncera un résultat pour une surface.

Polygone dans \mathbb{R}^2

On se place dans \mathbb{R}^2 . Soit un polygone convexe P , caractérisé par des sommets $\{x^j\}_{j=1}^p$ (figure 4.8). Par convention, on adopte le sens trigonométrique direct comme sens de parcours des sommets. L'arête A_j de P est le segment $[x^j, x^{j+1}]$, avec par convention $x^{p+1} = x^1$. On note ℓ_j la longueur de cette arête :

$$\ell_j = \|x^{j+1} - x^j\| ,$$

et on note w^j son orientation :

$$w^j = \frac{1}{\ell_j} (x^{j+1} - x^j) .$$

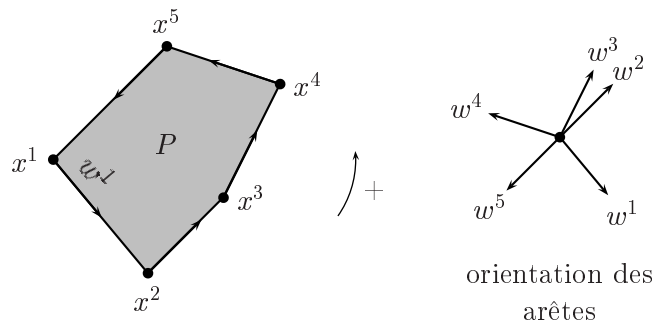


Fig. 4.8 – Polygone convexe dans \mathbb{R}^2 .

On note $\text{aire}(P)$ l'aire du polygone P . Enfin, on introduit les angles θ_j dans $[0, 2\pi[$ et $\gamma_{j \rightarrow j+1}$ dans $]0, \pi[$, orientés dans le sens trigonométrique direct (figure 4.9). Le premier représente l'angle entre la direction w^j et la direction w^1 , le second représente l'angle entre deux directions consécutives w^j et w^{j+1} , où par convention $w^{p+1} = w^1$. Par définition, on a donc :

$$\gamma_{j \rightarrow j+1} = \theta_{j+1} - \theta_j .$$

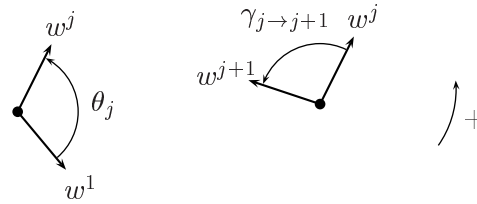


Fig. 4.9 – Notations pour les orientations des arêtes dans un polygone.

Le fait de prendre $\gamma_{j \rightarrow j+1} > 0$ pour tout j dans $\{1, \dots, p\}$, revient à supposer que deux arêtes consécutives de P ne sont pas colinéaires. De plus, cela témoigne de la convexité du polygone P , ce qui se traduit par un ordre sur les directions w^j , que l'on peut aussi exprimer de la manière suivante (figure 4.8) :

$$\forall j \in \{1, \dots, p-1\} , \theta_j < \theta_{j+1} .$$

La proposition suivante donne une estimation du nombre moyen de régions de \mathcal{S} traversées par un polygone convexe P .

Proposition 18

Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^2 traversées par un polygone convexe P est donné par le nombre suivant :

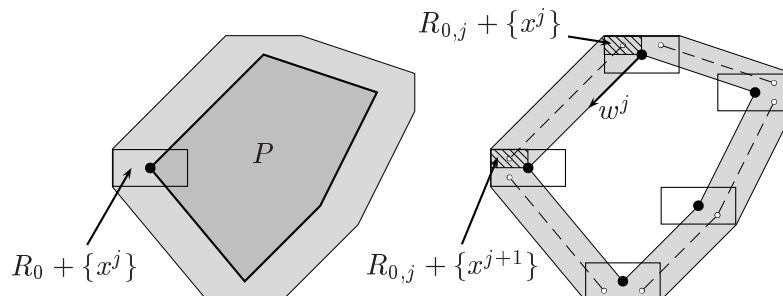
$$\bar{m}_P = 1 + \frac{1}{2} \sum_{j=1}^p \left(\sum_{i=0}^1 \frac{|w_i^j|}{\delta_i} \right) \ell_j + \frac{\text{aire}(P)}{\delta_0 \delta_1} , \quad (4.6)$$

où w_i^j est la coordonnée dans la direction i du vecteur w^j .

□

Preuve Pour évaluer \bar{m}_P on part de la définition 7, à savoir que le nombre moyen recherché correspond au rapport des aires de la somme de Minkowski $R_0 + P$ et d'une région de \mathcal{S} . On remarque tout d'abord que l'aire de l'ensemble $R_0 + P$ se décompose en deux parties : l'aire de P et l'aire de la couronne autour de P :

$$\text{aire}(R_0 + P) = \text{aire}(P) + \text{aire}((R_0 + P) \setminus P) .$$



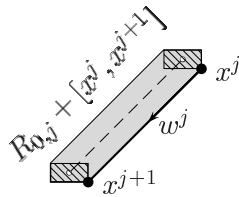
Pour évaluer l'aire de la couronne autour de P , il suffit de remarquer qu'elle se décompose par rapport aux arêtes de P de la manière suivante :

$$\text{aire}((R_0 + P) \setminus P) = \sum_{j=1}^p (\text{aire}(R_{0,j} + [x^j, x^{j+1}]) - \alpha_{j \rightarrow j+1}) , \tag{4.7}$$

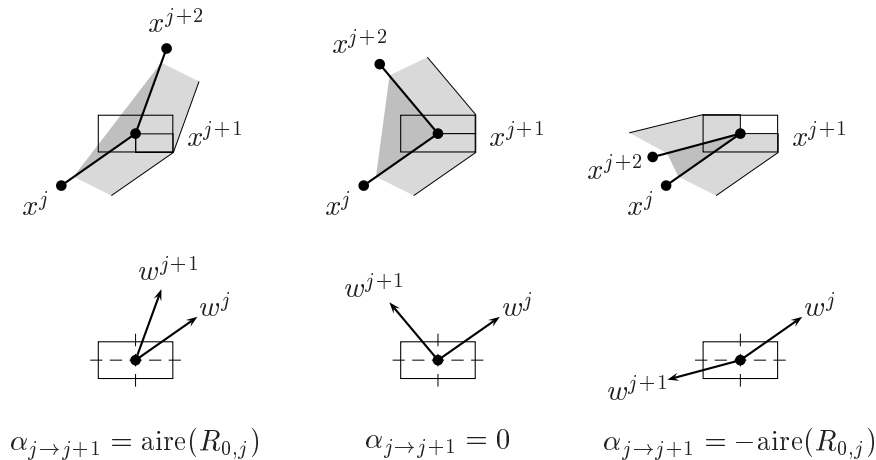
où $R_{0,j}$ est un quart de R_0 déterminé à partir de l'arête A_j . Le premier terme dans la somme est l'aire de la somme de Minkowski du domaine $R_{0,j}$ et de l'arête j :

$$\text{aire}(R_{0,j} + [x^j, x^{j+1}]) = \text{aire}(R_{0,j}) + \frac{1}{2} \left(\sum_{i=0}^1 \delta_{i+1} |w_i^j| \right) \ell_j , \tag{4.8}$$

où δ_{i+1} doit être compris comme $\delta_{(i+1) \bmod 2}$.



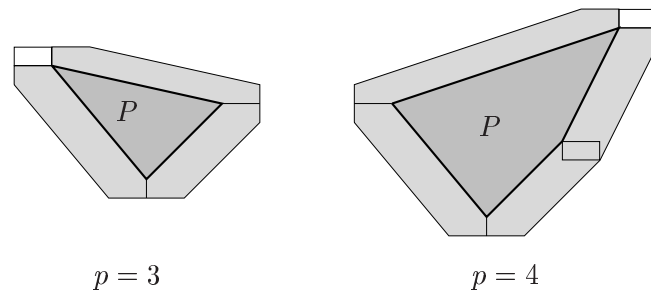
Le second terme dans la somme de l'expression 4.7 représente l'aire algébrique qu'il faut retrancher. Cette aire dépend de la configuration des deux arêtes consécutives considérées. Le dessin suivant donne l'expression de l'aire $\alpha_{j \rightarrow j+1}$ en fonction des configurations envisageables pour deux arêtes j et $j + 1$. Par convexité du polygone P , il y en a trois envisageables (l'angle $\gamma_{j \rightarrow j+1}$ est strictement inférieur à π).



Il ne reste plus qu'à dénombrer sur toutes les arêtes le nombre de fois où l'on retranche $\text{aire}(R_{0,j})$. On remarque la somme des $\alpha_{j \rightarrow j+1}$ sur l'ensemble des arêtes est constante pour un nombre d'arêtes p fixé. En effet, il suffit d'observer la répartition des orientations des arêtes dans les quadrangles $[0, \pi/2[$, $[\pi/2, \pi[$, $[\pi, 3\pi/2[$ et $[3\pi/2, 2\pi[$, et on rappelle que par convexité de P les directions des arêtes sont ordonnées. De plus l'aire de $R_{0,j}$ est le quart de l'aire de R_0 , pour tout j , d'où l'expression suivante :

$$\sum_{j=1}^p \alpha_{j \rightarrow j+1} = (p - 4) \text{aire}(R_{0,j}) = \frac{p - 4}{4} \text{aire}(R_0) . \tag{4.9}$$

Le facteur $p - 4$ peut être déterminé par une simple récurrence sur le nombre d'arêtes.



En injectant 4.8 et 4.9 dans 4.7, on obtient l'expression de l'aire de $R_0 + P$:

$$\text{aire}(R_0 + P) = \text{aire}(P) + \delta_0 \delta_1 + \frac{1}{2} \sum_{j=1}^p \left(\sum_{i=0}^1 \delta_{i+1} |w_i^j| \right) \ell_j .$$

Pour trouver l'expression proposée, il ne reste plus qu'à diviser l'aire de $R_0 + P$ par l'aire du rectangle R_0 . ■

Si l'on considère un polygone non convexe (figure 4.10), alors les notations introduites pour un polygone convexe restent valables, sauf pour l'angle $\gamma_{j \rightarrow j+1}$ qui est dans l'intervalle $]-\pi, 0[\cup]0, \pi[$, où l'intervalle $]-\pi, 0[$ correspond à une configuration concave d'arêtes successives.

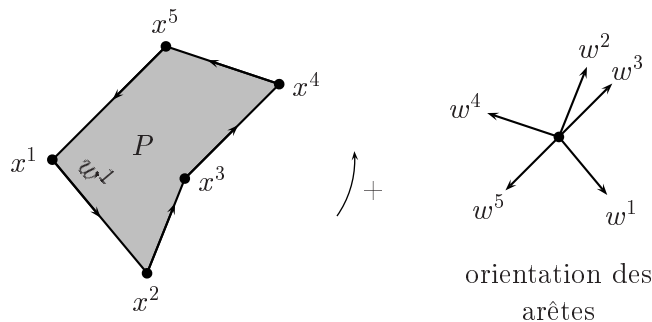


Fig. 4.10 – Polygone non convexe dans \mathbb{R}^2 .

La recherche du nombre moyen de régions d'une subdivision uniforme \mathcal{S} traversées par un polygone non convexe P est un problème difficile. Le corollaire suivant donne une majoration de ce nombre.

Corollaire 6

Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^2 traversées par un polygone P est majoré de la manière suivante :

$$\bar{m}_P \leq 1 + \frac{1}{2} \sum_{j=1}^p \left(\sum_{i=0}^1 \frac{|w_i^j|}{\delta_i} \right) \ell_j + \frac{\text{aire}(P)}{\delta_0 \delta_1} .$$

□

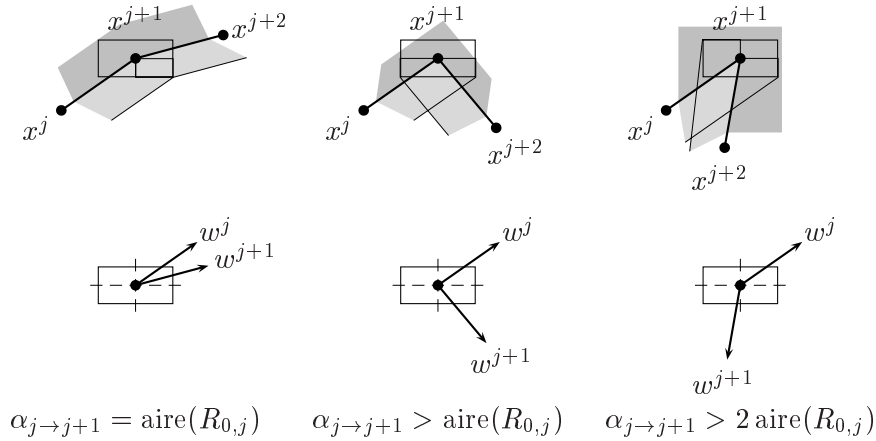
Preuve On raisonne de manière analogue à la démonstration de la proposition 18. Pour ce faire, on commence par décomposer l'aire de l'ensemble $R_0 + P$ en deux parties :

$$\text{aire}(R_0 + P) = \text{aire}(P) + \sum_{j=1}^p \text{aire}(R_{0,j} + [x^j, x^{j+1}]) - \sum_{j=1}^p \alpha_{j \rightarrow j+1} .$$

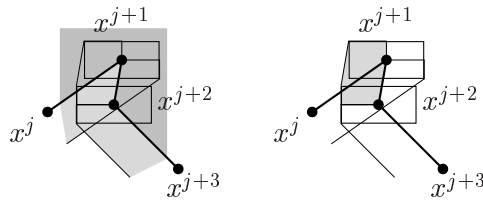
L'expression du deuxième terme est donné dans la démonstration précédente, seul le troisième terme est différent. Pour le calcul de l'aire algébrique $\alpha_{j \rightarrow j+1}$, il faut différencier les deux configurations d'arêtes envisageables. Si c'est une configuration convexe, c'est-à-dire $\gamma_{j \rightarrow j+1} \in]0, \pi[$, alors on est ramené aux trois cas de la démonstration précédente. Sinon, c'est une configuration concave, c'est à dire $\gamma_{j \rightarrow j+1} \in]-\pi, 0[$, et dans ce cas on remarque que :

$$\alpha_{j \rightarrow j+1} \geq \text{aire}(R_{0,j}) .$$

Pour s'en convaincre, il suffit d'observer les différentes configurations concaves. De manière analogue au cas convexe, on peut considérer trois cas.



Cependant, on ne perdra pas de vue qu'il y a en fait beaucoup plus de cas si l'on prend en compte la longueur des arêtes. En effet, si la longueur d'une arête est au plus de l'ordre de la taille d'une région, alors la couronne autour de P n'est pas nécessairement influencée par cette arête (dans le cas convexe ce problème ne se pose pas). Et dans ce cas, il faut envisager les configurations avec plus de deux arêtes successives. Mais, pour la majoration que l'on veut atteindre, ce n'est pas nécessaire, car ce type d'arêtes n'entrent pas en compte dans l'aire de la couronne autour de P , et par conséquent contribue à majorer le nombre recherché.



Ensuite, il reste à ajouter les contributions de toutes les arêtes. Chaque configuration d'arêtes successives correspond à une somme d'aires $\alpha_{j \rightarrow j+1}$. On a donc la majoration suivante :

$$\sum_{j=1}^p \alpha_{j \rightarrow j+1} \geq (p - 4) \text{aire}(R_{0,j}) = \frac{p - 4}{4} \text{aire}(R_0) .$$

D'où la majoration proposée. ■

Polygone plan dans \mathbb{R}^3

On se place dans \mathbb{R}^3 . Soit un polygone plan convexe P , caractérisé par des sommets $\{x^j\}_{j=1}^p$. On adopte les mêmes notations et les mêmes conventions qu'au paragraphe précédent, en se ramenant au plan contenant le polygone pour les conventions de parcours et les notions d'angles (figure 4.11). On notera Π_u le plan contenant le polygone P , où u est un vecteur normal unitaire à P .

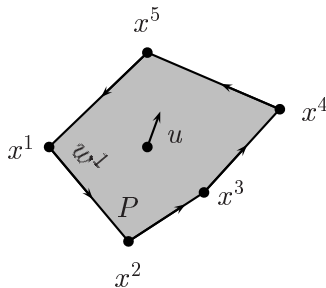


Fig. 4.11 – Notations pour un polygone dans \mathbb{R}^3 .

La proposition suivante donne une estimation du nombre moyen de régions de \mathcal{S} traversées par un polygone convexe P .

Proposition 19

Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^3 traversées par un polygone convexe P est donné par le nombre suivant :

$$\overline{m}_P = 1 + \frac{1}{2} \sum_{j=1}^p \left(\sum_{i=0}^2 \frac{|w_i^j|}{\delta_i} \right) \ell_j + \left(\sum_{i=0}^2 \delta_i |u_i| \right) \frac{\text{aire}(P)}{\delta_0 \delta_1 \delta_2}, \quad (4.10)$$

où w_i^j et u_i sont les coordonnées dans la direction i des vecteurs w^j et u .

□

Preuve Pour évaluer \overline{m}_P , on raisonne de manière analogue au cas \mathbb{R}^2 , c'est-à-dire à partir de la définition 7. Tout d'abord on écrit que le nombre moyen recherché correspond au rapport des volumes de la somme de Minkowski $R_0 + P$ et d'une région de \mathcal{S} . Pour calculer le volume de l'ensemble $R_0 + P$ on le décompose en deux volumes : le volume dû à P et le volume de la couronne autour de P :

$$\text{vol}(R_0 + P) = \text{vol}(C_{R_0}(P)) + \text{vol}((R_0 + P) \setminus C_{R_0}(P)),$$

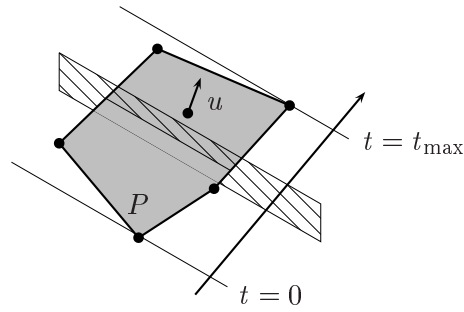
où C_{R_0} est l'opérateur constructif d'extrusion lié à R_0 . Le volume dû à P correspond au volume de l'extrusion de P dans une direction induite par l'orientation de Π_u (plan contenant P) et de longueur l'épaisseur de R_0 dans la direction u :

$$e_u = \sum_{i=0}^2 \delta_i |u_i|.$$

Pour évaluer le volume dû à P , il suffit de remarquer que l'on peut l'écrire sous la forme :

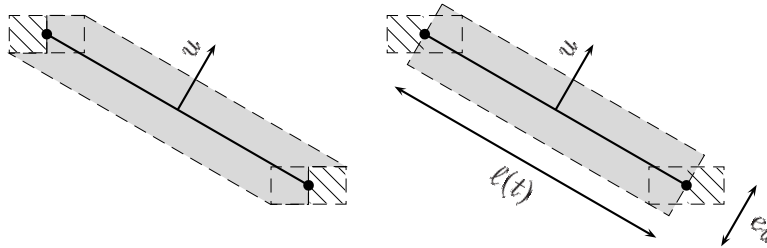
$$\text{vol}(C_{R_0}(P)) = \int_0^{t_{\max}} a(t) dt,$$

où $a(t)$ est l'aire de la section de partie de $R_0 + P$ due à P dans un plan perpendiculaire au plan Π_u placé à l'abscisse t .



Or, l'aire $a(t)$ s'écrit comme le produit de la longueur $\ell(t)$ de la section du polygone P par l'épaisseur de R_0 par rapport à la direction u :

$$a(t) = e_u \ell(t) .$$



D'où l'expression du volume recherché :

$$\begin{aligned} \text{vol}(C_{R_0}(P)) &= e_u \int_0^{t_{\max}} \ell(t) dt \\ &= e_u \text{aire}(P) . \end{aligned}$$

Le volume de la couronne autour de P se calcule de manière similaire au cas \mathbb{R}^2 , à la différence près que le volume de la somme de Minkowski du parallélépipède $R_{0,j}$ (un quart de R_0) et de l'arête A_j s'écrit de la manière suivante :

$$\text{vol}(R_{0,j} + [x^j, x^{j+1}]) = \text{vol}(R_{0,j}) + \frac{1}{2} \left(\sum_{i=0}^2 \delta_{i+1} \delta_{i+2} |w_i^j| \right) \ell_j ,$$

où δ_{i+1} et δ_{i+2} doivent être compris comme $\delta_{(i+1) \bmod 3}$ et $\delta_{(i+2) \bmod 3}$. Ensuite, on effectue la somme sur toutes les arêtes, en tenant compte des volumes à rajouter ou à retrancher. Le volume de la couronne autour de P s'écrit alors :

$$\text{vol}((R_0 + P) \setminus C_{R_0}(P)) = \text{vol}(R_0) + \frac{1}{2} \sum_{j=1}^p \left(\sum_{i=0}^2 \delta_{i+1} \delta_{i+2} |w_i^j| \right) \ell_j .$$

En ajoutant les deux volumes calculés puis en divisant par le volume de R_0 on obtient l'expression proposée. ■

Pour montrer que la proposition 19 est une généralisation du résultat de Cleary et Wyvill [18], commençons par supposer que la taille des régions de la subdivision uniforme considérée est identique dans toutes les directions :

$$\delta_i = \delta .$$

Ainsi 4.10 s'écrit de la manière suivante :

$$\bar{m}_P = 1 + \frac{1}{2\delta} \sum_{j=1}^p \left(\sum_{i=0}^2 |w_i^j| \right) \ell_j + \frac{1}{\delta^2} \left(\sum_{i=0}^2 |u_i| \right) \text{aire}(P) .$$

La moyenne par rapport à l'ensemble des directions de la sphère unité de ce nombre s'écrit :

$$\langle \bar{m}_P \rangle = \frac{1}{16\pi^2} \int_{\mathcal{S}^2} \int_{\mathcal{S}^2} \bar{m}_P \, d\sigma(w) \, d\sigma(u) .$$

L'intégrale précédente est séparable. On remarque, d'une part, que :

$$\int_{\mathcal{S}^2} \sum_{i=0}^2 |u_i| \, d\sigma(u) = 6\pi ,$$

et, d'autre part, que pour tout $j \in \{1, \dots, p\}$:

$$\int_{\mathcal{S}^2} \sum_{i=0}^2 |w_i^j| \, d\sigma(w) = 6\pi .$$

Finalement, le nombre moyen de régions d'une subdivision uniforme traversées par un polygone convexe P s'écrit :

$$\langle \bar{m}_P \rangle = 1 + \frac{3}{4\delta} \text{perim}(P) + \frac{3}{2\delta^2} \text{aire}(P) ,$$

où $\text{perim}(P)$ est le périmètre du polygone P , c'est-à-dire la somme des longueurs des arêtes de P . L'expression précédente est le résultat donné dans [18]. L'expression 4.10 est donc bien une généralisation de ce résultat.

Le triangle est un polygone plan convexe particulièrement intéressant. En effet, dans bien des cas, une surface quelconque est approchée par un ensemble de triangles, à l'aide d'un procédé de triangulation. Il est donc intéressant de formuler l'expression du nombre de régions traversées par un triangle.

Soit T un triangle. Soit \mathcal{S} une subdivision uniforme de \mathbb{R}^3 . La proposition 19 donne le nombre moyen de régions traversées par un T . Pour simplifier l'écriture de l'équation 4.10, introduisons les deux coefficients réels suivants, caractéristiques du triangle T et de la subdivision \mathcal{S} . Soit $\mu_j(w)$ un coefficient associé à l'arête A_j de T et défini de la manière suivante :

$$\mu_j(w) = \frac{1}{2} \sum_{i=0}^2 \frac{|w_i^j|}{\delta_i} .$$

Soit $\nu(u)$ un coefficient associé au triangle et défini de la manière suivante :

$$\nu(u) = \frac{1}{\delta_0 \delta_1 \delta_2} \sum_{i=0}^2 \delta_i |u_i| .$$

Le nombre de régions de la subdivision \mathcal{S} traversées par le triangle T est donné par l'équation 4.10, qui peut alors s'écrire de la manière suivante :

$$\bar{m}_T = 1 + \sum_{j=1}^3 \mu_j(w) \ell_j + \nu(u) \text{aire}(T) . \quad (4.11)$$

De manière analogue au cas \mathbb{R}^2 , on adapte les notations et on donne une majoration du nombre moyen de régions d'une subdivision uniforme \mathcal{S} traversées par un polygone non convexe P .

Corollaire 7

Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^3 traversées par un polygone P est majoré de la manière suivante :

$$\bar{m}_P \leq \bar{m}_P^+ = 1 + \frac{1}{2} \sum_{j=1}^p \left(\sum_{i=0}^2 \frac{|w_i^j|}{\delta_i} \right) \ell_j + \left(\sum_{i=0}^2 \delta_i |u_i| \right) \frac{\text{aire}(P)}{\delta_0 \delta_1 \delta_2}.$$

□

Preuve Le même raisonnement que le cas \mathbb{R}^2 (corollaire 6) peut être utilisé ici. Il suffit de remarquer que les configurations concaves d'arêtes successives tendent à diminuer le volume de la couronne autour de P .

■

Surface affine par morceaux

Soit Σ une surface simple continue, constituée par un ensemble fini de polygones plans (figure 4.12) :

$$\Sigma = \bigcup_{i=1}^{n_{\text{poly}}} P_i.$$

Σ est une surface affine par morceaux, sans auto-intersection.

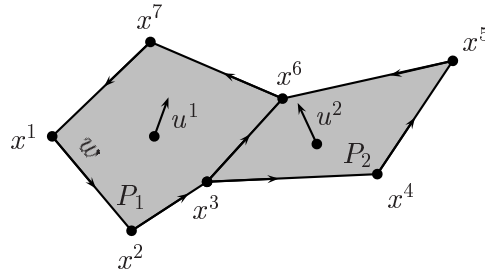


Fig. 4.12 – Notations pour une surface constituée de polygones plans.

Les notations et les conventions pour le placement et l'orientation de Σ sont induites par les polygones P_i . On note $\{x^j\}_{j=1}^p$ l'ensemble des sommets de Σ et $A(\Sigma)$ l'ensemble des arêtes de Σ :

$$A(\Sigma) = \{A = [x^j, x^{j'}], j, j' \in \{1, \dots, p\}, A \text{ arête de } \Sigma\}.$$

Pour différencier les arêtes de bord de Σ des autres, on introduit les deux sous-ensembles de $A(\Sigma)$ suivants :

$$\begin{cases} A_{\text{bord}}(\Sigma) &= \{A \in A(\Sigma) \text{ tel que } A \subset \partial\Sigma\} \\ A_{\text{in}}(\Sigma) &= A(\Sigma) \setminus A_{\text{bord}}(\Sigma) \end{cases}.$$

La proposition suivante donne une majoration du nombre de régions de \mathcal{S} traversées par une surface affine par morceaux.

Proposition 20

Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^3 traversées par une surface Σ simple continue, constituée d'un ensemble fini de polygones plans $\{P_i\}_{i=1}^{n_{\text{poly}}}$ est majoré de la manière suivante :

$$\overline{m}_\Sigma \leq 2 + \sum_{i=1}^{n_{\text{poly}}} (\overline{m}_{P_i}^+ - 1) - \sum_{A \in A_{\text{in}}(\Sigma)} (\overline{m}_A - 1) , \quad (4.12)$$

où $\overline{m}_{P_i}^+$ est le majorant du nombre moyen de régions traversées par le polygone P_i donné par le corollaire 7, \overline{m}_A est le nombre moyen de régions traversées par l'arête A donné par la proposition 16 et $A_{\text{in}}(\Sigma)$ est l'ensemble des arêtes intérieures à Σ .

□

Preuve Pour majorer le nombre \overline{m}_Σ , on part de la définition 7. On cherche à calculer le volume de la somme de Minkowski $R_0 + \Sigma$. Cet ensemble est constitué de deux parties, l'une issue de la surface Σ et l'autre due à la couronne autour de Σ . On le décompose sur l'ensemble des polygones :

$$R_0 + \Sigma = R_0 + \bigcup_{i=1}^{n_{\text{poly}}} P_i = \bigcup_{i=1}^{n_{\text{poly}}} (R_0 + P_i) .$$

Le lemme 3 page 147 donne une expression du volume recherché :

$$\text{vol}(R_0 + \Sigma) = \sum_{i=1}^{n_{\text{poly}}} \text{vol}(R_0 + P_i) - \sum_{i=1}^{n_{\text{poly}}} \text{vol} \left(\bigcup_{j>i} ((R_0 + P_i) \cap (R_0 + P_j)) \right) .$$

Le lemme 3 peut être à nouveau utilisé pour donner une expression du second terme dans la somme :

$$\begin{aligned} \text{vol} \left(\bigcup_{j>i} ((R_0 + P_i) \cap (R_0 + P_j)) \right) &= \sum_{j>i} \text{vol}((R_0 + P_i) \cap (R_0 + P_j)) \\ &\quad - \sum_{j>i} \text{vol} \left(\bigcup_{k>j} ((R_0 + P_i) \cap (R_0 + P_j) \cap (R_0 + P_k)) \right) . \end{aligned}$$

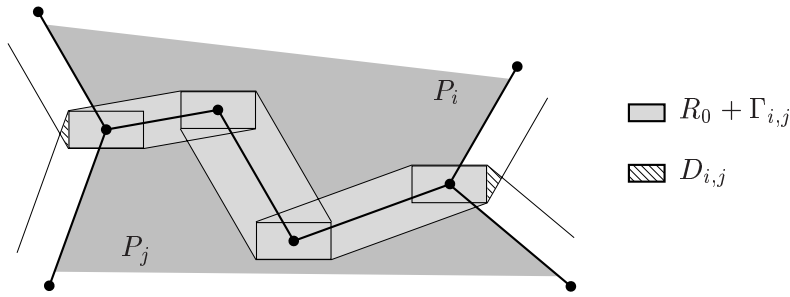
L'expression du volume de $R_0 + \Sigma$ s'écrit alors de la manière suivante :

$$\begin{aligned} \text{vol}(R_0 + \Sigma) &= \sum_{i=1}^{n_{\text{poly}}} \text{vol}(R_0 + P_i) - \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol}((R_0 + P_i) \cap (R_0 + P_j)) \\ &\quad + \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol} \left(\bigcup_{k>j} ((R_0 + P_i) \cap (R_0 + P_j) \cap (R_0 + P_k)) \right) . \end{aligned} \quad (4.13)$$

Il s'agit maintenant d'analyser de plus près les deux derniers termes de l'expression précédente. Le second terme représente la somme des volumes des intersections deux à deux entre les polygones $\{P_i\}_i$. Ce terme contient donc les volumes relatifs aux arêtes intérieures de Σ , ce qui s'écrit :

$$\sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol}((R_0 + P_i) \cap (R_0 + P_j)) = \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol}(R_0 + \Gamma_{i,j}) + \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol}(D_{i,j}) , \quad (4.14)$$

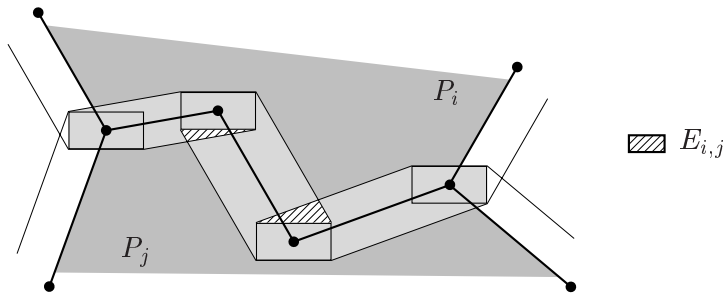
où $\Gamma_{i,j}$ est l'ensemble des arêtes communes aux polygones P_i et P_j et $D_{i,j}$ sont les ensembles représentant les éventuelles parties non comptées lors de la décomposition.



Le premier terme de l'expression 4.14 contient la somme des volumes sur les arêtes de $\Gamma_{i,j}$, ce qui s'écrit :

$$\text{vol}(R_0 + \Gamma_{i,j}) = \sum_{A \in A(\Gamma_{i,j})} \text{vol}(R_0 + A) - \sum_{S \in S_{\text{in}}(\Gamma_{i,j})} \text{vol}(R_0) - \text{vol}(E_{i,j}) , \quad (4.15)$$

où $A(\Gamma_{i,j})$ est l'ensemble des arêtes de $\Gamma_{i,j}$, $S_{\text{in}}(\Gamma_{i,j})$ est l'ensemble des sommets intérieurs à $\Gamma_{i,j}$ (ce sont les sommets qui ont exactement deux arêtes incidentes) et $E_{i,j}$ est l'ensemble des parties comptées en trop lors de la décomposition.



Le troisième terme dans 4.13 représente la somme des volumes des intersections trois à trois entre les polygones $\Gamma_{i,j}$. Ce terme contient donc les volumes relatifs aux sommets intérieurs à Σ qui sont communs à au moins trois polygones, donc qui ont au moins trois arêtes intérieures incidentes, ce qui s'écrit :

$$\sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol} \left(\bigcup_{k>j} ((R_0 + P_i) \cap (R_0 + P_j) \cap (R_0 + P_k)) \right) = \sum_{S \in S_{\text{in},3}(\Gamma_{i,j})} \text{vol}(R_0) + \text{vol}(F) , \quad (4.16)$$

où $S_{\text{in},3}(\Sigma)$ représente l'ensemble des sommets intérieurs à Σ qui ont au moins trois arêtes incidentes et F est l'ensemble des parties non comptées lors de cette décomposition. En injectant 4.14, 4.15 et 4.16 dans 4.13, on obtient une nouvelle expression du volume de l'ensemble $R_0 + \Sigma$:

$$\begin{aligned} \text{vol}(R_0 + \Sigma) &= \sum_{i=1}^{n_{\text{poly}}} \text{vol}(R_0 + P_i) - \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \sum_{A \in A(\Gamma_{i,j})} \text{vol}(R_0 + A) + \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \sum_{S \in S_{\text{in}}(\Gamma_{i,j})} \text{vol}(R_0) \\ &+ \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol}(E_{i,j}) - \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol}(D_{i,j}) + \sum_{S \in S_{\text{in},3}(\Gamma_{i,j})} \text{vol}(R_0) + \text{vol}(F) . \end{aligned} \quad (4.17)$$

Le second terme de 4.17 n'est autre que la somme sur l'ensemble des arêtes intérieures à Σ :

$$\sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \sum_{A \in A(\Gamma_{i,j})} \text{vol}(R_0 + A) = \sum_{A \in A_{\text{in}}(\Sigma)} \text{vol}(R_0 + A) . \quad (4.18)$$

Les troisième et sixième termes de 4.17 se regroupent en la somme des sommets intérieurs à Σ :

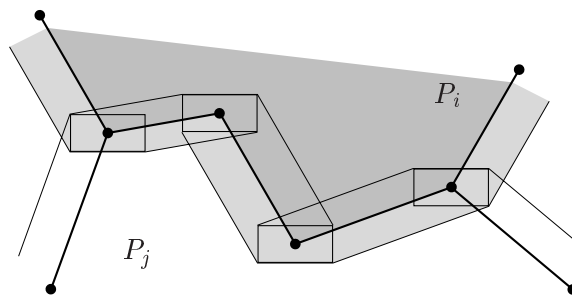
$$\sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \sum_{S \in S_{\text{in}}(\Gamma_{i,j})} \text{vol}(R_0) + \sum_{S \in S_{\text{in},3}(\Gamma_{i,j})} \text{vol}(R_0) = \sum_{S \in S_{\text{in}}(\Sigma)} \text{vol}(R_0) , \quad (4.19)$$

où $S_{\text{in}}(\Sigma)$ représente l'ensemble des sommets intérieurs à Σ . Il ne reste plus qu'à regrouper les termes restants. Les premier et quatrième termes se regroupent à l'aide du corollaire 7 considéré sur l'ensemble des polygones $\{P_i\}_i$. Celui-ci s'écrit de la manière suivante :

$$\sum_{i=1}^{n_{\text{poly}}} \text{vol}(R_0 + P_i) \leq \text{vol}(R_0) \sum_{i=1}^{n_{\text{poly}}} \bar{m}_{P_i}^+ .$$

Le majorant peut être affiné. En effet, il suffit de remarquer que les ensembles $E_{i,j}$ introduits précédemment sont comptés en double lors de la majoration de la somme des volumes des $R_0 + P_i$. Ainsi, on peut regrouper les premier et quatrième termes de 4.17 :

$$\sum_{i=1}^{n_{\text{poly}}} \text{vol}(R_0 + P_i) + \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol}(E_{i,j}) \leq \text{vol}(R_0) \sum_{i=1}^{n_{\text{poly}}} \bar{m}_{P_i}^+ . \quad (4.20)$$



Enfin, il ne reste plus qu'à remarquer que les ensembles $D_{i,j}$ sont issus d'intersections entre deux polygones alors que F est issu d'intersections entre trois polygones. On obtient donc l'inégalité suivante :

$$- \sum_{i=1}^{n_{\text{poly}}} \sum_{j>i} \text{vol}(D_{i,j}) + \text{vol}(F) \leq 0 . \quad (4.21)$$

En injectant les expressions 4.18, 4.19, 4.20 et 4.21 dans 4.17, puis en divisant le tout par le volume de R_0 on obtient la majoration suivante :

$$\bar{m}_{\Sigma} \leq \sum_{i=1}^{n_{\text{poly}}} \bar{m}_{P_i}^+ - \sum_{A \in A_{\text{in}}(\Sigma)} \bar{m}_A + n_{\text{sommets, in}} ,$$

où $n_{\text{sommets, in}}$ est le cardinal de l'ensemble $S_{\text{in}}(\Sigma)$, c'est-à-dire le nombre de sommets situés à l'intérieur de Σ . Le membre de droite dans l'expression précédente peut s'écrire sous la forme :

$$\sum_{i=1}^{n_{\text{poly}}} (\bar{m}_{P_i}^+ - 1) - \sum_{A \in A_{\text{in}}(\Sigma)} (\bar{m}_A - 1) + n_{\text{poly}} - n_{\text{aretes, in}} + n_{\text{sommets, in}} .$$

La relation d'Euler caractérisant la topologie de la surface Σ est majorée de la manière suivante (Kinsey [58]) :

$$n_{\text{poly}} - (n_{\text{aretes, in}} + n_{\text{aretes, ex}}) + (n_{\text{sommets, in}} + n_{\text{sommets, ex}}) \leq 2 ,$$

où $n_{\text{arêtes, ex}}$ et $n_{\text{sommets, ex}}$ sont respectivement les nombres d'arêtes et de sommets situés sur le bord de Σ . Puisque $n_{\text{arêtes, ex}} - n_{\text{sommets, ex}} = 0$, on a finalement l'expression suivante :

$$n_{\text{poly}} - n_{\text{arêtes, in}} + n_{\text{sommets, in}} \leq 2 .$$

D'où le résultat proposé. ■

4.2.4 Boîte englobante

Soit $\mathcal{S} = \{R_k\}_{k \in \mathbb{Z}}$ une subdivision uniforme de \mathbb{R}^3 . Soit B une boîte englobante, parallèle aux axes (figure 4.13).

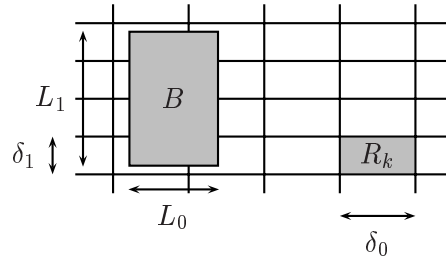


Fig. 4.13 – Notations pour l'évaluation du nombre de régions traversées par une boîte englobante (parallèle aux axes).

La proposition suivante caractérise le nombre moyen de régions de \mathcal{S} traversées par une boîte englobante B parallèle aux axes.

Proposition 21

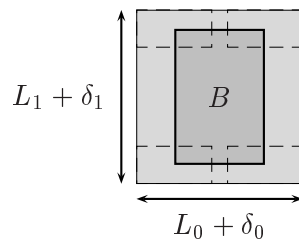
Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^3 traversées par une boîte englobante B parallèle aux axes, caractérisée par des longueurs L_i , est donné par le nombre suivant :

$$\overline{m}_B = \frac{1}{\delta_0 \delta_1 \delta_2} \prod_{i=0}^2 (L_i + \delta_i) . \quad (4.22)$$

□

Preuve Partant de la définition 7, il suffit de calculer le volume de $R_0 + B$:

$$\text{vol}(R_0 + B) = \prod_{i=0}^2 (L_i + \delta_i) .$$



Pour trouver l'expression proposée, il ne reste plus qu'à diviser le volume de $R_0 + B$ par le volume de R_0 . ■

La proposition précédente est limitée à des boîtes englobantes particulières. Généralisons ce résultat à des boîtes quelconques. Soit B une boîte englobante quelconque (figure 4.14).

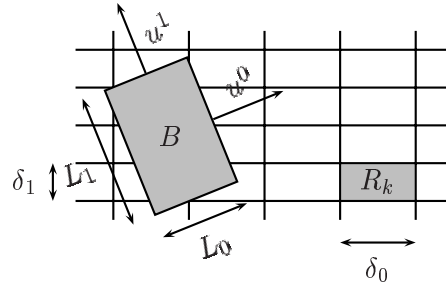


Fig. 4.14 – Notations pour l'évaluation du nombre de régions traversées par une boîte englobante.

La proposition suivante caractérise le nombre moyen de régions de \mathcal{S} traversées par une boîte englobante B non nécessairement parallèle aux axes.

Proposition 22

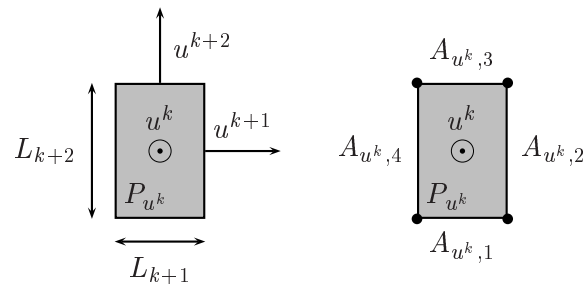
Le nombre moyen de régions d'une subdivision uniforme \mathcal{S} de \mathbb{R}^3 traversées par une boîte englobante B , caractérisée par des longueurs L_k et des vecteurs normaux unitaires u^k , est donné par le nombre suivant :

$$\bar{m}_B = \sum_{k=0}^2 \left(\left(\sum_{i=0}^2 \frac{|u_i^k|}{\delta_i} \right) L_k + \left(\sum_{i=0}^2 \delta_i |u_i^k| \right) \frac{L_{k+1} L_{k+2}}{\delta_0 \delta_1 \delta_2} \right) + \frac{L_0 L_1 L_2}{\delta_0 \delta_1 \delta_2}, \quad (4.23)$$

où u_i^k est la coordonnée dans la direction i du vecteur u^k , et L_{k+1} et L_{k+2} doivent être compris comme $L_{(k+1) \bmod 3}$ et $L_{(k+2) \bmod 3}$.

□

Preuve On part de la définition 7, et on cherche à calculer le volume de l'ensemble $R_0 + B$. Pour ce faire, on décompose en deux parties : le volume de la boîte B et le volume de la couronne autour de B . Pour ce dernier, on utilise les résultats obtenus sur les polygones convexes dans la démonstration de la proposition 19. Soit P_{u^k} une face de B , et soit P_{-u^k} la face opposée. Soient $A_{u^k, j}$, pour $j \in \{1, \dots, 4\}$, les arêtes de la face P_{u^k} .



Pour calculer le volume de la couronne autour de B , il suffit de décomposer par rapport aux faces de B . Pour ce faire, on ajoute les volumes des ensembles $R_0 + P_{u^k}$ et $R_0 + P_{-u^k}$, en retranchant la partie en intersection avec B , puis on retranche les parties comptées en double au niveau des arêtes. Tout d'abord, on remarque que :

$$\text{vol}(R_0 + P_{-u^k}) = \text{vol}(R_0 + P_{u^k}) .$$

Les arêtes parallèles vérifient le même type le relation, et on note :

$$\begin{cases} \text{vol}(R_0 + A_{\pm u^k,1}) = \text{vol}(R_0 + A_{\pm u^k,3}) = \text{vol}(R_0 + A_{u^{k+1}}) \\ \text{vol}(R_0 + A_{\pm u^k,2}) = \text{vol}(R_0 + A_{\pm u^k,4}) = \text{vol}(R_0 + A_{u^{k+2}}) \end{cases} .$$

Le volume de la couronne autour de B s'écrit alors de la manière suivante :

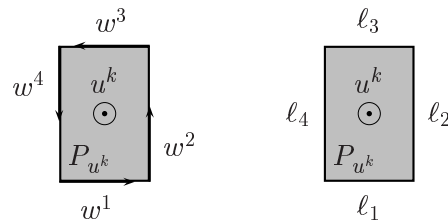
$$\sum_{k=0}^2 (\text{vol}(R_0 + P_{u^k}) - \text{vol}(R_0 + A_{u^k})) . \quad (4.24)$$

Il ne reste donc plus qu'à calculer chaque volume de l'expression précédente. Se référant à la démonstration de la proposition 19, le volume de l'ensemble $R_0 + P_{u^k}$ se décompose en le volume du P_{u^k} et le volume de la couronne autour de P_{u^k} :

$$\text{vol}(R_0 + P_{u^k}) = \left(\sum_{i=0}^2 \delta_i |u_i^k| \right) \text{aire}(P_{u^k}) + \sum_{j=1}^4 \left(\text{vol}(R_0, j) + \frac{1}{2} \left(\sum_{i=0}^2 \delta_{i+1} \delta_{i+2} |w_i^j| \right) \ell_j \right) ,$$

où l'indice j se réfère à l'arête j du polygone P_{u^k} , et, d'après les notations, on pose :

$$\begin{cases} w^1 = \pm u^{k+1} = -w^3 \\ w^2 = \pm u^{k+2} = -w^4 \end{cases} \quad \text{et} \quad \begin{cases} \ell_1 = L_{k+1} = \ell_3 \\ \ell_2 = L_{k+2} = \ell_4 \end{cases} .$$



Finalement on a :

$$\begin{aligned} \text{vol}(R_0 + P_{u^k}) &= \left(\sum_{i=0}^2 \delta_i |u_i^k| \right) L_{k+1} L_{k+2} + \text{vol}(R_0) \\ &\quad + \left(\sum_{i=0}^2 \delta_{i+1} \delta_{i+2} |u_i^{k+1}| \right) L_{k+1} + \left(\sum_{i=0}^2 \delta_{i+1} \delta_{i+2} |u_i^{k+2}| \right) L_{k+2} . \end{aligned}$$

Toujours par référence à la démonstration de la proposition 19, le volume de l'ensemble $R_0 + A_{u^k}$ est le volume du parallélépipède résultant :

$$\text{vol}(R_0 + A_{u^k}) = \text{vol}(R_0) + \left(\sum_{i=0}^2 \delta_{i+1} \delta_{i+2} |u_i^k| \right) L_k .$$

En regroupant les deux expressions obtenues dans 4.24, on obtient le volume de la couronne autour de B :

$$\sum_{k=0}^2 \left(\left(\sum_{i=0}^2 \delta_i |u_i^k| \right) L_{k+1} L_{k+2} + \left(\sum_{i=0}^2 \delta_{i+1} \delta_{i+2} |u_i^k| \right) L_k \right) .$$

Pour obtenir l'expression proposée, il suffit de diviser le volume de $R_0 + B$ par le volume de R_0 . ■

La proposition 22 est bien une généralisation de la proposition 21. Pour s'en convaincre, considérons une boîte englobante B parallèle aux axes, c'est à dire dont les vecteurs normaux unitaires sont :

$$\begin{cases} u^0 = (1, 0, 0) \\ u^1 = (0, 1, 0) \\ u^2 = (0, 0, 1) \end{cases} .$$

L'expression 4.23 s'écrit alors :

$$\begin{aligned} \overline{m}_B &= \sum_{k=0}^2 \left(\frac{L_k}{\delta_k} + \delta_k \frac{L_{k+1} L_{k+2}}{\delta_0 \delta_1 \delta_2} \right) + \frac{L_0 L_1 L_2}{\delta_0 \delta_1 \delta_2} \\ &= \frac{1}{\delta_0 \delta_1 \delta_2} \left(\sum_{k=0}^2 (\delta_{k+1} \delta_{k+2} L_k + \delta_k L_{k+1} L_{k+2}) + L_0 L_1 L_2 \right) , \end{aligned}$$

ce qui s'écrit en développant la somme et en regroupant les termes :

$$\overline{m}_B = \frac{(L_0 + \delta_0)(L_1 + \delta_1)(L_2 + \delta_2)}{\delta_0 \delta_1 \delta_2} .$$

On retrouve bien l'expression 4.22.

4.3 Complexité en mémoire

4.3.1 Caractérisation

De manière analogue à la définition 6 page 96 où l'on caractérise la liste des indices des objets qui possèdent une intersection avec une région donnée, la définition suivante caractérise la liste des indices des régions qui ont une intersection avec un objet donné.

Définition 8

Soit R un parallélépipède. Soit $\mathcal{S} = \{R_k\}_{k=1}^n$ une subdivision de R . Soit O un objet. On note $L_{\Upsilon(O)}$ la liste des indices des régions de \mathcal{S} qui possèdent une intersection avec l'objet O :

$$L_{\Upsilon(O)} = \{k \in \{1, \dots, n\}, \Upsilon(O) \cap R_k \neq \emptyset\} ,$$

où $\Upsilon(O)$ représente l'objet considéré pour l'évaluation de l'intersection avec une région R_k . □

La proposition suivante donne une expression de la complexité en mémoire relative à une scène géométrique et une subdivision uniforme données. Bien entendu, seul les objets susceptibles d'interagir avec des rayons sont considérés, c'est-à-dire les objets surfaciques et les objets volumiques.

Proposition 23

Soit R un voxel. Soit \mathcal{S} une subdivision de R . Soient O^1, \dots, O^N des objets. La complexité en mémoire de la localisation des objets O^1, \dots, O^N dans \mathcal{S} s'écrit :

$$M_{\mathcal{S}, O^1, \dots, O^N} = \sum_{i=1}^N (M_{O^i} + \text{card}(L_{\Upsilon(O^i)}) M_{v_o}) . \quad (4.25)$$

où M_{O^i} est la complexité en mémoire relative à l'objet O^i et M_{v_o} est la complexité en mémoire d'un enregistrement de lien voxel-objet. □

Preuve Soit R un voxel. Soit $\mathcal{S} = \{R_k\}_{k=1}^n$ une subdivision de R . Soient O^1, \dots, O^N des objets (surfaciques ou volumiques). La complexité en mémoire de la localisation des N objets dans R consiste en le stockage des objets O^1, \dots, O^N et le stockage des liens voxel-objet, c'est-à-dire des listes $\{L_{R_k}\}_{k=1}^n$ associées aux régions $\{R_k\}_{k=1}^n$. La complexité en mémoire s'écrit donc de la manière suivante :

$$M_{\mathcal{S}, O^1, \dots, O^N} = \sum_{i=1}^N M_{O^i} + \sum_{k=1}^n M_{L_{R_k}} .$$

Le premier terme est donné au deuxième chapitre, pour chaque modèle géométrique considéré. Explicitons le second terme. Un lien voxel-objet est décrit par un enregistrement constitué d'un ensemble de paramètres représentant les différentes informations nécessaires au repérage des objets dans les régions R_k (nombre d'objets, indice de chaque objet lié, indices précisant des sous-parties des objets, indices de régions pour la gestion de la hiérarchie, etc). Supposons que la taille d'un enregistrement de lien voxel-objet M_{vo} est identique pour chaque région. Ainsi, la complexité en mémoire pour les liens voxel-objet s'écrit :

$$\sum_{k=1}^n M_{L_{R_k}} = \left(\sum_{k=1}^n \text{card}(L_{R_k}) \right) M_{\text{vo}} .$$

La somme du nombre d'objets par région sur \mathcal{S} est égale à la somme du nombre de régions par objets sur O^1, \dots, O^N :

$$\sum_{k=1}^n \text{card}(L_{R_k}) = \sum_{i=1}^N \text{card}(L_{\Upsilon(O^i)}) .$$

D'où l'expression de la complexité en mémoire proposée. ■

4.3.2 Estimation

La complexité en mémoire est caractérisée par la proposition 23. Dans l'expression 4.25 intervient le nombre de régions d'une subdivision uniforme traversées par un objet. Dans la section 4.2 il est montré que l'évaluation exacte de ce nombre n'est pas envisageable, ce qui a impliqué une estimation moyenne. On introduit alors une complexité moyenne approchée $M'_{\mathcal{S}, O^1, \dots, O^N}$:

$$\boxed{M'_{\mathcal{S}, O^1, \dots, O^N} = \sum_{i=1}^N (M_{O^i} + \bar{m}_{\Upsilon(O^i)} M_{\text{vo}})} , \quad (4.26)$$

où $\bar{m}_{\Upsilon(O^i)}$ est le nombre de régions de \mathcal{S} traversées par l'objet O^i , au sens de la localisation envisagée pour cet objet.

La localisation d'un objet O dans une subdivision uniforme consiste à rechercher les régions de la subdivision qui ont une intersection avec un objet lié à O , noté $\Upsilon(O)$. L'idéal est de considérer la surface optique associée à O , c'est-à-dire $\Sigma(O)$, au sens de la définition 5 page 96. Cependant, la complexité géométrique de certains objets ne permet pas d'avoir une localisation aussi précise qu'on le désire. C'est pourquoi nous présentons dans ce paragraphe deux localisations. La première est basée sur une boîte englobante, la deuxième sur la surface optique.

Localisation par rapport à une boîte englobante

Dans ce paragraphe on s'intéresse à la localisation d'un objet par rapport à une boîte englobante associée (figure 4.15). Ce qui revient à chercher les régions de la subdivision uniforme considérée qui

ont une intersection avec la boîte englobante considérée. Le paragraphe 4.2.4 traite de ce problème. La complexité en mémoire d'une telle localisation est donnée par la proposition 21.

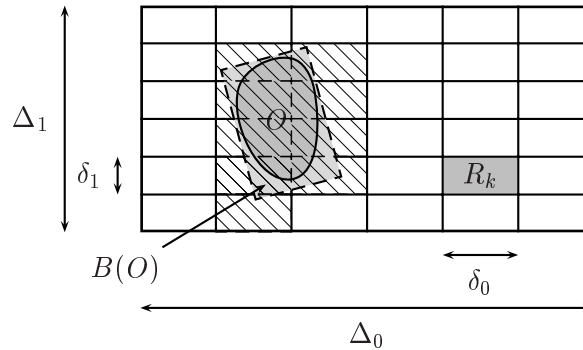
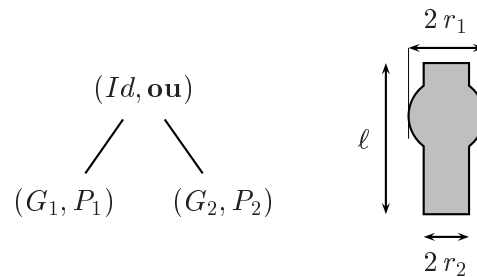


Fig. 4.15 – Localisation par rapport à une boîte englobante associée à un objet.

Exemple 9

Reprenons l'exemple 2 page 52. Soit R un cube de côté 1. Soit une scène géométrique incluse dans ce cube et contenant un objet CSG O construit par union d'une sphère P_1 , caractérisée par un rayon r_1 , et d'un cylindre P_2 , caractérisé par un rayon $r_2 < r_1$ et une longueur $\ell > 2r_1$. Un opérateur de translation G_1 est appliqué à la sphère tandis qu'un opérateur de rotation G_2 est appliqué au cylindre. Aucun opérateur géométrique n'est appliqué sur l'objet résultant de l'union.



L'exemple 2 précise que la complexité en mémoire pour l'objet O est :

$$M_O = 12 .$$

Soit \mathcal{S} une subdivision uniforme de R telle que $\delta_i = \delta$ dans chaque direction. Autrement dit, le nombre de subdivisions dans chaque direction est :

$$n_i = n^{1/3} .$$

Enfin, soit $B(O)$ une boîte englobante minimale de O , parallèle aux axes. Par construction, sa longueur dans une direction est ℓ et sa longueur dans les deux autres directions est $2r_1$. Le nombre moyen de régions de \mathcal{S} traversées par $B(O)$ est donné par l'expression 4.22 :

$$\bar{m}_{B(O)} = n_i^3 \left(2r_1 + \frac{1}{n_i} \right)^2 \left(\ell + \frac{1}{n_i} \right) .$$

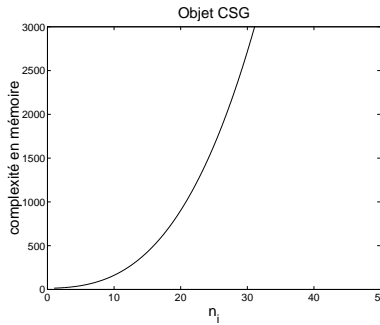
La complexité en mémoire approchée de la localisation s'écrit donc :

$$M'_{\mathcal{S},O} = 12 + n_i^3 \left(2r_1 + \frac{1}{n_i} \right)^2 \left(\ell + \frac{1}{n_i} \right) M_{\text{vo}} .$$

Pour illustrer la complexité en mémoire approchée, fixons les valeurs suivantes pour les paramètres à considérer :

$$\left\{ \begin{array}{l} r_1 = 0.2 \\ r_2 = 0.1 \\ \ell = 0.5 \\ M_{\text{vo}} = 1 \end{array} \right. .$$

Le graphe suivant illustre la complexité en mémoire en fonction du nombre de subdivisions.



□

Localisation par rapport à la surface optique

Dans ce paragraphe on s'intéresse à la localisation d'un objet par rapport à la surface optique associée (figure 4.16). En d'autres termes, et par définition de la surface optique, on s'intéresse aux intersections entre une surface et les régions d'une subdivision. Le paragraphe 4.2.3 traite de ce problème. Il suffit que la surface optique puisse être approchée par une surface affine par morceaux. La complexité en mémoire d'une telle subdivision est estimée à l'aide de la proposition 20.

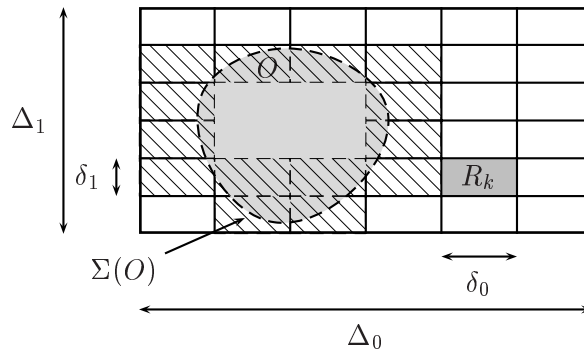


Fig. 4.16 – Localisation par rapport à la surface optique associée à un objet.

Exemple 10

Reprenons l'exemple 6 page 78. Soit R un cube de côté 1. Soit une scène géométrique incluse dans ce cube et contenant un objet BRep O composé de n_{face} faces triangulaires. L'exemple 6 précise que la complexité en mémoire pour l'objet O est :

$$M_O = 54 n_{\text{face}} .$$

Soit \mathcal{S} une subdivision uniforme de R telle que $\delta_i = \delta$ dans chaque direction. Autrement dit, le nombre de subdivisions dans chaque direction est :

$$n_i = \frac{1}{\delta_i} = n^{1/3} .$$

L'expression 4.12 majore le nombre moyen de régions de la subdivision \mathcal{S} traversées par l'objet O . Pour l'évaluer, il faut distinguer les triangles et les arêtes intérieures à O . Dans un premier temps, commençons par approcher l'expression 4.11 du nombre de régions traversées par un triangle T_j en ne tenant pas compte de l'orientation des arêtes de O , autrement dit en calculant la moyenne de \overline{m}_{T_j} par rapport à la variable w :

$$\langle \overline{m}_{T_j} \rangle = 1 + \frac{3}{4} \text{perim}(T_j) n_i + \frac{3}{2} \text{aire}(T_j) n_i^2 ,$$

où $\text{perim}(T)$ est le périmètre du triangle T . Dans un second temps, évaluons le nombre moyen de régions de \mathcal{S} traversées par les arêtes intérieures à O à l'aide de l'expression 4.3. Ne tenons pas compte de l'orientation des arêtes, pour être cohérent avec l'hypothèse faite sur les triangles. Ainsi, pour une arête A_k de longueur ℓ_k , le nombre moyen recherché s'écrit :

$$\langle \overline{m}_{A_k} \rangle = 1 + \frac{3}{2} \ell_k n_i .$$

Les nombres considérés des deux cotés de l'expression 4.12 étant positifs, on peut écrire la moyenne par rapport à l'ensemble des directions, et on a la majoration suivante :

$$\langle \overline{m}_O \rangle \leq 2 + \sum_{j=1}^{n_{\text{face}}} (\overline{m}'_{T_j} - 1) - \sum_{k=1}^{n_{\text{arête,in}}} (\overline{m}'_{A_k} - 1) .$$

En développant la borne supérieure, on remarque qu'elle s'écrit sous la forme suivante :

$$2 + \frac{3}{2} \left(\frac{1}{2} \sum_{j=1}^{n_{\text{face}}} \text{perim}(T_j) - \sum_{k=1}^{n_{\text{arête,in}}} \ell_k \right) n_i + \frac{3}{2} \text{aire}(O) n_i^2 .$$

Or, l'expression de la somme des périmètres sur l'ensemble des triangles peut se mettre sous la forme suivante :

$$\sum_{j=1}^{n_{\text{face}}} \text{perim}(T_j) = \sum_{k=1}^{n_{\text{arête,ex}}} \ell_k + 2 \sum_{k=1}^{n_{\text{arête,in}}} \ell_k .$$

D'où une majoration du nombre de régions de \mathcal{S} traversées par l'objet O :

$$\langle \overline{m}_O \rangle \leq 2 + \frac{3}{4} \text{perim}(O) n_i + \frac{3}{2} \text{aire}(O) n_i^2 .$$

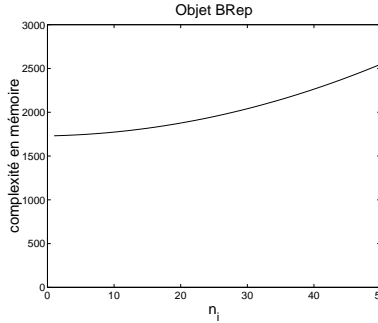
La complexité en mémoire approchée, et moyennée par rapport à l'ensemble des directions, de la localisation de l'objet O dans \mathcal{S} est alors majorée de la manière suivante :

$$\langle M'_{\mathcal{S},O} \rangle \leq 54 n_{\text{face}} + \left(2 + \frac{3}{4} \text{perim}(O) n_i + \frac{3}{2} \text{aire}(O) n_i^2 \right) M_{\text{vo}} .$$

Pour illustrer la complexité en mémoire approchée, fixons les valeurs suivantes pour les paramètres à considérer :

$$\left\{ \begin{array}{l} n_{\text{face}} = 32 \\ \text{perim}(O) = 1.8 \\ \text{aire}(O) = 0.2 \\ M_{\text{vo}} = 1 \end{array} \right. .$$

Le graphe suivant illustre la borne supérieure de la complexité en mémoire en fonction du nombre de subdivisions.



□

4.4 Complexité en temps

4.4.1 Caractérisation

La proposition suivante donne une expression de la complexité en temps de recherche de la première intersection entre un rayon et une scène géométrique plongée dans une subdivision uniforme donnée (algorithme 10 page 113, algorithme 11 page 115 et algorithme 12 page 116). Seuls les objets surfaciques et volumiques sont considérés.

Proposition 24

Soit $R_{x,w}$ un rayon. Soit R un voxel. Soit \mathcal{S} une subdivision de R . Soient O^1, \dots, O^N des objets. La complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et les objets O^1, \dots, O^N s'écrit de la manière suivante :

$$T_{\mathcal{S}, O^1, \dots, O^N}(x, w) = T_{\text{psv}} + m(x, w) T_v + \sum_{i \in I(x, w)} T_{O^i}(x, w), \quad (4.27)$$

où T_{psv} est la complexité en temps de recherche de la première région de \mathcal{S} traversée par $R_{x,w}$, $m(x, w)$ est le nombre de régions de \mathcal{S} traversées par $R_{x,w}$, T_v est la complexité en temps de parcours d'une région de \mathcal{S} , $I(x, w)$ est l'ensemble des indices des objets rencontrés lors du parcours de $R_{x,w}$ dans \mathcal{S} :

$$I(x, w) = \{i \in \{1, \dots, N\}, \Upsilon(O^i) \cap R_{k_j} \neq \emptyset, j \in \{1, \dots, m(x, w)\}\},$$

et $T_{O^i}(x, w)$ est la complexité en temps de recherche de la première intersection entre le rayon $R_{x,w}$ et l'objet O^i .

□

Preuve Soit $R_{x,w}$ un rayon. Soit R un voxel. Soit \mathcal{S} une subdivision de R . Soient O^1, \dots, O^N des objets (surfacciques ou volumiques). La complexité en temps de recherche de la première intersection entre le rayon considéré et la scène géométrique considérée (algorithme 12 page 116) se décompose en deux parties.

Tout d'abord, il est nécessaire de rechercher la première région de \mathcal{S} traversée par le rayon (algorithme 10 page 113). La complexité en temps associée est indépendante du rayon considéré puisque cette recherche consiste en un calcul d'intersection entre le rayon $R_{x,w}$ et le voxel R (calcul indépendant du rayon) et trois divisions tronquées (l'intersection entre le rayon et le voxel existe toujours). Cette complexité en temps est T_{psv} .

Ensuite, le rayon parcourt les régions jusqu'à trouver une intersection avec un objet O^i ou sortir du voxel R . La complexité en temps du parcours de $m(x, w)$ régions de \mathcal{S} (algorithme 11 page 115) s'écrit :

$$m(x, w) T_v .$$

La complexité en temps de calcul des intersections (un seul calcul d'intersection est pris en compte par objet rencontré) s'écrit :

$$\sum_{i \in I(x, w)} T_{O^i}(x, w) .$$

D'où le résultat proposé. ■

La proposition 24 donne une expression de la complexité en temps difficilement exploitable, et ce à cause du troisième terme de l'équation 4.27. Le corollaire suivant donne une expression plus abordable de la complexité en temps dans un cas particulier.

Corollaire 8

Si les objets O^1, \dots, O^N sont tels que :

$$T_{O^i}(x, w) = T_O(x, w) , \forall i \in \{1, \dots, N\} , \forall (x, w) \in \mathcal{R} ,$$

alors la complexité en temps de recherche de la première intersection entre le rayon $R_{x, w}$ et les objets O^1, \dots, O^N s'écrit :

$$T_{\mathcal{S}, O^1, \dots, O^N}(x, w) = T_{\text{psv}} + m(x, w) T_v + N(x, w) T_O(x, w) , \quad (4.28)$$

où $N(x, w)$ est le cardinal de l'ensemble $I(x, w)$ et représente le nombre d'objets rencontrés lors du parcours du rayon $R_{x, w}$ dans \mathcal{S} . □

Preuve Il suffit de remplacer le troisième terme dans l'équation 4.27. ■

L'expression de la complexité en temps donnée dans le corollaire 8 peut être vue comme une approximation de la complexité en temps donnée par l'équation 4.27 dans le cas d'objets quelconques. Pour ce faire, il suffit de considérer une complexité en temps moyennée sur l'ensemble des objets :

$$T_O(x, w) = \frac{1}{N} \sum_{i=1}^N T_{O^i}(x, w) .$$

La complexité en temps de recherche de la première intersection entre un rayon $R_{x, w}$ et les objets O^1, \dots, O^N est approchée de la manière suivante :

$$T_{\mathcal{S}, O^1, \dots, O^N}(x, w) \approx T_{\text{psv}} + m(x, w) T_v + N(x, w) T_O(x, w) .$$

Cependant, ce qui nous intéresse n'est pas une complexité en temps pour un rayon $R_{x, w}$ particulier, mais plutôt une complexité moyenne en temps¹ sur l'ensemble des rayons issus de \mathcal{R} . Pour écrire cette complexité moyenne en temps, il suffit d'écrire la complexité moyenne en temps pour chaque terme de l'équation 4.27. Les deux premiers termes ne posent pas de problème. Par contre, le troisième terme est plus difficile à manipuler. La proposition suivante donne une expression de la complexité moyenne en temps dans un cas particulier.

¹Voir définition 1 page 36.

Proposition 25

Soit R un voxel. Soit \mathcal{S} une subdivision de R . Soient O^1, \dots, O^N des objets dont les complexités en temps T_{O^i} sont telles que :

$$T_{O^i}(x, w) = T_O, \forall i \in \{1, \dots, N\}, \forall (x, w) \in \mathcal{R}.$$

La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et les objets O^1, \dots, O^N s'écrit de la manière suivante :

$$\overline{T}_{\mathcal{S}, O^1, \dots, O^N} = T_{\text{psv}} + \overline{m} T_v + \overline{N} T_O. \quad (4.29)$$

□

Preuve Il suffit d'appliquer la définition 1 de la complexité moyenne en temps (équation 2.1 page 36) à l'équation 4.28. Le troisième terme ne pose pas de problème car T_O est constante. ■

L'expression de la complexité moyenne en temps donnée dans la proposition 25 peut être considérée comme une approximation de la complexité moyenne en temps dans le cas d'objets quelconques. Pour ce faire, il suffit de considérer une complexité en temps constante et moyennée sur l'ensemble des objets considérés :

$$T_O = \frac{1}{N} \sum_{i=1}^N \overline{T}_{O^i}.$$

La complexité moyenne en temps de recherche de la première intersection entre un rayon issu de \mathcal{R} et les objets O^1, \dots, O^N est approchée de la manière suivante :

$$\overline{T}_{\mathcal{S}, O^1, \dots, O^N} \approx T_{\text{psv}} + \overline{m} T_v + \overline{N} T_O.$$

Pour expliciter la complexité moyenne en temps, il reste à évaluer \overline{m} , le nombre moyen de régions traversées, et \overline{N} , le nombre moyen d'objets rencontrés.

4.4.2 Ensemble d'objets

Pour l'évaluation de la complexité moyenne en temps, les objets considérés doivent satisfaire certaines conditions. D'un point de vue géométrique, chaque objet doit être tel que sa surface optique associée soit continue, dérivable par morceaux, sans auto-intersection. De plus, lorsqu'on considère un ensemble d'objets O^1, \dots, O^N , ceux-ci ne doivent pas s'interpénétrer, seules certaines relations d'adjacences sont admises. La définition suivante caractérise un ensemble d'objets admissible pour les simulations envisagées, et donc pour les paragraphes suivants.

Définition 9

Deux objets de type surfacique ou volumique sont dits optiquement cohérents si leur intersection est soit l'ensemble vide, soit un ensemble de points disjoints, soit un ensemble de points décrivant un ou plusieurs objets linéiques disjoints. De plus, deux objets de type volumique sont dit optiquement cohérents si leur intersection est un ensemble de points décrivant un ou plusieurs objets surfaciques disjoints.

Un ensemble d'objets O^1, \dots, O^N est dit optiquement cohérent si les objets sont deux à deux optiquement cohérents.

□

Par exemple, deux sphères tangentes sont optiquement cohérentes, l'ensemble des polygones plans d'une surface affine par morceaux est optiquement cohérent puisque les intersections deux à deux sont des arêtes, deux cubes adjacents par une face sont optiquement cohérents. Par contre, deux triangles se chevauchant ne sont pas optiquement cohérents.

4.4.3 Ensemble de rayons

Pour caractériser la complexité moyenne en temps il est nécessaire de se donner l'ensemble des rayons issus de \mathcal{R} . Connaître a priori tous les rayons ayant une existence pour un système optique donné est utopique. Dans ce paragraphe, on cherche donc à définir un ensemble de rayons qui est représentatif des simulations effectuées sur le système optique pris en compte.

Pour ce faire, on souhaite s'affranchir des propriétés physiques du système optique considéré. Cependant, on prendra en compte la propriété d'opacité des objets, qui, d'un point de vue géométrique, a une influence importante sur l'ensemble de rayons que l'on va considérer par la suite. Par définition, aucun rayon ne traverse un objet (volumique) opaque. On introduit alors les deux sous-ensembles de Σ suivants, où l'on rappelle que Σ est l'ensemble des points situés sur les surfaces optiques :

- Σ_o l'ensemble des points situés sur les contours des objets (volumiques) opaques,
- Σ_{no} l'ensemble complémentaire à Σ_o dans Σ .

On a donc la relation :

$$\Sigma = \Sigma_o \cup \Sigma_{no} .$$

Soit R un voxel dans le domaine de simulation X . Soient O^1, \dots, O^N des objets. On s'intéresse à l'ensemble des rayons qui traversent R , c'est pourquoi on introduit l'ensemble \mathcal{R}_R , qui est un sous-ensemble de \mathcal{R} :

$$\mathcal{R}_R = \{(x, w) \in \mathcal{R} \text{ tel que } R_{x,w} \cap R \neq \emptyset\} .$$

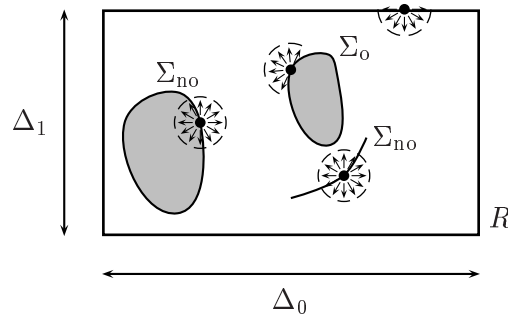
Pour avoir un ensemble de rayons le plus représentatif, indépendant des propriétés optiques des objets, hormis l'opacité, on prend en compte les surfaces optiques associées à l'ensemble des objets et toutes les directions de l'espace, autrement dit l'ensemble de rayons à considérer est le suivant :

$$\mathcal{R}_R = \mathcal{R}_{\Sigma_o} \cup \mathcal{R}_{\Sigma_{no}} \cup \mathcal{R}_{\partial R} , \tag{4.30}$$

où les ensembles \mathcal{R}_{Σ_o} et $\mathcal{R}_{\Sigma_{no}}$ représentent les rayons issus respectivement des objets opaques et des objets non opaques et $\mathcal{R}_{\partial R}$ représentent les rayons issus du bord du voxel R . Ils sont définis de la manière suivante :

$$\left\{ \begin{array}{l} \mathcal{R}_{\Sigma_o} = \{(x, w) \in \mathcal{R} \text{ tel que } x \in \Sigma_o \text{ et } w \in \Omega^+(x)\} \\ \mathcal{R}_{\Sigma_{no}} = \{(x, w) \in \mathcal{R} \text{ tel que } x \in \Sigma_{no} \text{ et } w \in \mathcal{S}^2\} \\ \mathcal{R}_{\partial R} = \{(x, w) \in \mathcal{R} \text{ tel que } x \in \partial R \text{ et } w \in \Omega^-(x)\} \end{array} \right. ,$$

où les sous-ensembles $\Omega^-(x)$ et $\Omega^+(x)$ de \mathcal{S}^2 sont caractérisés au paragraphe 1.1.3. Cette expression de l'ensemble des couples (x, w) représentatifs des rayons pris en compte signifie que l'on considère une distribution uniforme des rayons sur les objets surfaciques et sur les contours des objets volumiques, dans les directions strictement extérieures pour les objets opaques et dans toutes les directions de l'espace pour les objets non opaques, et sur les bords du voxel R , dans les directions strictement intérieures au voxel R (figure 4.17).

Fig. 4.17 – Définition de l'ensemble des rayons issus de \mathcal{R}_R .

On a vu au paragraphe 2.1.2 que l'ensemble \mathcal{R} est mesurable (au sens de la mesure produit $\varrho = \mu \times \sigma$). De plus, les ensembles \mathcal{R}_{Σ_o} , $\mathcal{R}_{\Sigma_{no}}$ et $\mathcal{R}_{\partial R}$ sont disjoints, donc l'ensemble \mathcal{R}_R est mesurable, et sa mesure s'exprime de la manière suivante :

$$\text{mes}(\mathcal{R}_R) = \text{mes}(\mathcal{R}_{\Sigma_o}) + \text{mes}(\mathcal{R}_{\Sigma_{no}}) + \text{mes}(\mathcal{R}_{\partial R}) .$$

La mesure de l'ensemble des couples (x, w) représentant les rayons issus de objets opaques s'écrit :

$$\text{mes}(\mathcal{R}_{\Sigma_o}) = \int_{\Sigma_o} \left(\int_{\Omega^+(x)} d\sigma(w) \right) d\mu(x) = 2\pi \text{aire}(\Sigma_o) .$$

De même, la mesure de l'ensemble des couples (x, w) représentant les rayons issus des objets non opaques s'écrit :

$$\text{mes}(\mathcal{R}_{\Sigma_{no}}) = \int_{\Sigma_{no}} \left(\int_{S^2} d\sigma(w) \right) d\mu(x) = 4\pi \text{aire}(\Sigma_{no}) .$$

Enfin, la mesure de l'ensemble des couples (x, w) représentant les rayons issus du bord de la région R s'écrit :

$$\text{mes}(\mathcal{R}_{\partial R}) = \int_{\partial R} \left(\int_{\Omega^-(x)} d\sigma(w) \right) d\mu(x) = 2\pi \text{aire}(\partial R) .$$

Finalement, la mesure de l'ensemble \mathcal{R}_R s'écrit :

$$\text{mes}(\mathcal{R}_R) = 2\pi (\text{aire}(\Sigma_o) + 2\text{aire}(\Sigma_{no}) + \text{aire}(\partial R)) . \quad (4.31)$$

4.4.4 Longueur d'un rayon

Soit R un voxel. Soient O^1, \dots, O^N des objets et Σ l'ensemble des surfaces optiques (définition 5 page 96) associées à ces objets. Soit $R_{x,w}$ un rayon traversant R . La longueur du rayon $R_{x,w}$ est la fonction distance aux surfaces, donnée par l'équation 1.12 page 21, considérée sur $\mathcal{R}_R = \mathcal{R}_{\Sigma_o} \cup \mathcal{R}_{\Sigma_{no}} \cup \mathcal{R}_{\partial R}$:

$$\begin{aligned} \nu_{\Sigma \cup \partial R} : \quad \mathcal{R}_R &\longrightarrow [0, d_R] \\ (x, w) &\longmapsto \nu_{\Sigma \cup \partial R}(x, w) \end{aligned} ,$$

où d_R est la longueur de la diagonale du voxel R :

$$d_R = \sqrt{\sum_{i=0}^2 \Delta_i^2} .$$

Essentiellement pour des raisons pratiques d'écriture, on introduit la fonction longueur $\ell_{\Sigma \cup \partial R}$, définie sur $(\Sigma \cup \partial R) \times \mathcal{S}^2$, à valeurs dans $[0, d_R]$:

$$\ell_{\Sigma \cup \partial R}(x, w) = \begin{cases} \nu_{\Sigma \cup \partial R}(x, w) & \text{si } (x, w) \in \mathcal{R}_R \\ 0 & \text{sinon} \end{cases} . \quad (4.32)$$

La fonction $\ell_{\Sigma \cup \partial R}$ n'est autre que le prolongement de $\nu_{\Sigma \cup \partial R}$ sur $(\Sigma \cup \partial R) \times \mathcal{S}^2$. Ne connaissant pas a priori $\nu_{\Sigma \cup \partial R}$, on ne peut pas donner une expression de la longueur d'un rayon particulier. Par contre, il est possible d'évaluer la longueur moyenne d'un rayon issu de \mathcal{R}_R , dont la définition est la suivante.

Définition 10

La longueur moyenne d'un rayon issu de \mathcal{R}_R est donnée par l'expression suivante :

$$\bar{\nu}_{\Sigma \cup \partial R} = \frac{1}{\text{mes}(\mathcal{R}_R)} \int_{\mathcal{R}_R} \nu_{\Sigma \cup \partial R}(x, w) d\mu(x) d\sigma(w) . \quad (4.33)$$

□

L'équation 4.33 est valable puisque l'intégrale a bien un sens. En effet, la fonction $\nu_{\Sigma \cup \partial R}$ est borné, l'ensemble \mathcal{R}_R est borné, donc $\nu_{\Sigma \cup \partial R}$ est dans l'espace métrique $L^1(\mathcal{R}_R)$. Par définition de la fonction $\ell_{\Sigma \cup \partial R}$, on peut écrire la longueur moyenne d'un rayon sous la forme suivante :

$$\bar{\nu}_{\Sigma \cup \partial R} = \frac{1}{\text{mes}(\mathcal{R}_R)} \int_{\Sigma \cup \partial R} \int_{\mathcal{S}^2} \ell_{\Sigma \cup \partial R}(x, w) d\mu(x) d\sigma(w) . \quad (4.34)$$

Le problème qui se pose alors est de calculer la longueur moyenne d'un rayon issu de \mathcal{R}_R . Pour ce faire, il suffit de calculer l'intégrale dans l'expression 4.33. On remarquera que ce calcul peut s'effectuer en décomposant l'intégrale double par rapport à chaque variable et en commençant par intégrer par rapport à la variable d'espace x puis en terminant par rapport à la variable de direction w . Le résultat ne fait alors intervenir que les caractéristiques géométriques du voxel R et des objets O^1, \dots, O^N , en tenant compte de la propriété d'opacité.

De manière similaire au paragraphe 1.1.3 où l'on décompose la sphère unité en trois sous-ensembles, on introduit la décomposition de ∂R suivante. Pour une direction w fixée dans \mathcal{S}^2 , on note ∂R_w^- , ∂R_w^0 et ∂R_w^+ les trois sous-ensembles de points de ∂R suivants :

$$\begin{cases} \partial R_w^- = \{x \in \partial R \text{ tel que } \langle w, u(x) \rangle < 0\} \\ \partial R_w^0 = \{x \in \partial R \text{ tel que } \langle w, u(x) \rangle = 0\} \\ \partial R_w^+ = \{x \in \partial R \text{ tel que } \langle w, u(x) \rangle > 0\} \end{cases} ,$$

où $u(x)$ est le vecteur normal unitaire au point x du bord du voxel R , orienté vers l'extérieur de R (figure 4.18). On a la relation suivante :

$$\partial R = \partial R_w^- \cup \partial R_w^0 \cup \partial R_w^+ .$$

L'ensemble ∂R_w^0 caractérise les points de tangence entre les droites d'orientation w et le bord du voxel R . Il est vide pour presque toutes les directions de \mathcal{S}^2 , sauf celles qui sont parallèles à une des faces de R , c'est-à-dire les directions w telles qu'au moins une composante soit nulle.

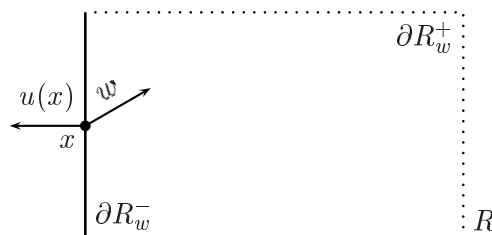


Fig. 4.18 – Définition de ∂R_w^- et ∂R_w^+ ($\partial R_w^0 = \emptyset$ sur cet exemple).

On adopte des notations analogues pour l'ensemble des surfaces optiques associées aux objets opaques :

$$\begin{cases} \Sigma_{o,w}^- = \{x \in \Sigma_o \text{ tel que } \langle w, u(x) \rangle < 0\} \\ \Sigma_{o,w}^0 = \{x \in \Sigma_o \text{ tel que } \langle w, u(x) \rangle = 0\} \\ \Sigma_{o,w}^+ = \{x \in \Sigma_o \text{ tel que } \langle w, u(x) \rangle > 0\} \end{cases} ,$$

où $u(x)$ est le vecteur normal unitaire au point x de la surface Σ_o , orienté vers l'extérieur de l'objet (volumique) associé. On a la relation suivante :

$$\Sigma_o = \Sigma_{o,w}^- \cup \Sigma_{o,w}^0 \cup \Sigma_{o,w}^+ .$$

L'ensemble $\Sigma_{o,w}^0$ caractérise les points de tangence entre les droites d'orientation w et le bord de la surface Σ_o .

La proposition suivante donne l'expression de la longueur moyenne recherchée.

Proposition 26

Soit R un voxel. Soient O^1, \dots, O^N des objets optiquement cohérents. La longueur moyenne d'un rayon issu de \mathcal{R}_R est donnée par l'expression suivante :

$$\bar{\nu}_{\Sigma \cup \partial R} = \frac{2 \left(\text{vol}(R) - \sum_{i=1}^N v_i \right)}{\text{aire}(\partial R) + \sum_{i=1}^N a_i} , \quad (4.35)$$

où v_i est défini par :

$$v_i = \begin{cases} \text{vol}(O^i) & \text{si } O^i \text{ est un objet volumique opaque} \\ 0 & \text{sinon} \end{cases} , \quad (4.36)$$

et a_i est défini par :

$$a_i = \begin{cases} \text{aire}(\partial O^i) & \text{si } O^i \text{ est un objet volumique opaque} \\ 2 \text{aire}(\Sigma(O^i)) & \text{sinon} \end{cases} , \quad (4.37)$$

où $\Sigma(O^i)$ est la surface optique associée à l'objet O^i .

□

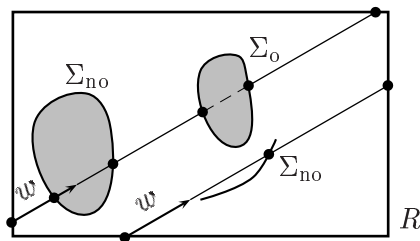
Preuve On part de l'expression 4.33 de la définition 10 qui donne la longueur moyenne :

$$\bar{\nu}_{\Sigma \cup \partial R} = \frac{1}{\text{mes}(\mathcal{R}_R)} \int_{\mathcal{R}_R} \nu_{\Sigma \cup \partial R}(x, w) d\mu(x) d\sigma(w) .$$

Pour calculer l'intégrale dans cette expression, on décompose par rapport à chaque variable. Pour écrire l'intégration par rapport à la variable x , il faut faire attention au domaine de définition de la fonction $\nu_{\Sigma \cup \partial R}$. Une écriture directe serait délicate, c'est pourquoi on utilise le prolongement de la fonction $\nu_{\Sigma \cup \partial R}$ sur $(\Sigma \cup \partial R) \times \mathcal{S}^2$ pour utiliser un domaine d'intégration plus simple :

$$\begin{aligned} \int_{\mathcal{R}_R} \nu_{\Sigma \cup \partial R}(x, w) d\mu(x) d\sigma(w) &= \int_{(\Sigma \cup \partial R) \times \mathcal{S}^2} \ell_{\Sigma \cup \partial R}(x, w) d\mu(x) d\sigma(w) \\ &= \int_{\mathcal{S}^2} \left(\int_{\Sigma \cup \partial R} \ell_{\Sigma \cup \partial R}(x, w) d\mu(x) \right) d\sigma(w) . \end{aligned}$$

Soit w fixé dans \mathcal{S}^2 . On s'intéresse à l'intégrale par rapport à x dans l'expression précédente.



Sommer les longueurs des rayons issus de $\Sigma \cup \partial R$ est équivalent à sommer les longueurs des rayons issus de ∂R et interceptant uniquement ∂R en retranchant les intersections avec les objets volumiques opaques. Ce qui s'écrit de la manière suivante :

$$\int_{\Sigma \cup \partial R} \ell_{\Sigma \cup \partial R}(x, w) d\mu(x) = \int_{\partial R} \ell_{\partial R}(x, w) d\mu(x) - \int_{\Sigma_0} \sigma_{\Sigma_0}(x, w) d\mu(x) ,$$

où $\sigma_{\Sigma_0}(x, w)$ est la longueur de la corde issue du rayon $R_{x,w}$ et interceptant l'objet (volumique) O^i tel que $x \in \partial O^i$. On a donc :

$$\begin{aligned} \int_{\Sigma \cup \partial R} \ell_{\Sigma \cup \partial R}(x, w) d\mu(x) &= \int_{\partial R_w^-} \nu_{\partial R}(x, w) d\mu(x) - \int_{\Sigma_0^-} \nu_{\Sigma_0}(x, w) d\mu(x) \\ &= \text{vol}(R) - \sum_{i=1}^N v_i . \end{aligned}$$

Cette expression est indépendante de w . L'intégration par rapport à w est donc triviale :

$$\int_{\mathcal{R}_R} \nu_{\Sigma \cup \partial R}(x, w) d\mu(x) d\sigma(w) = 4\pi \left(\text{vol}(R) - \sum_{i=1}^N v_i \right) .$$

Il ne reste plus qu'à diviser par la mesure de l'ensemble \mathcal{R}_R donnée par l'équation 4.31 :

$$\begin{aligned} \text{mes}(\mathcal{R}_R) &= 2\pi (\text{aire}(\Sigma_0) + 2 \text{aire}(\Sigma_{\text{no}}) + \text{aire}(\partial R)) \\ &= 2\pi \left(\text{aire}(\partial R) + \sum_{i=1}^N a_i \right) . \end{aligned}$$

D'où l'expression proposée pour la longueur moyenne d'un rayon. ■

Exemple 11

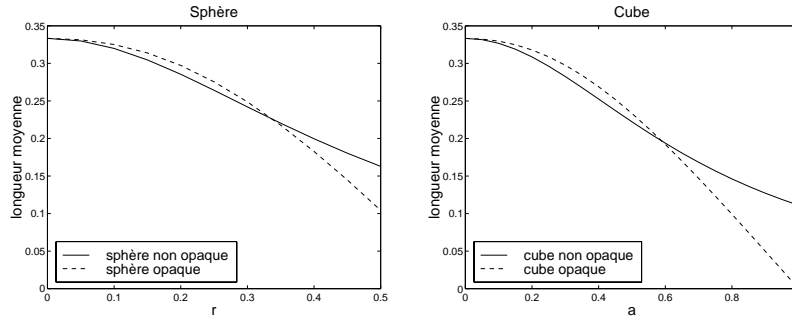
Soit R un cube de côté 1. Soit une scène géométrique incluse dans ce cube et contenant un seul objet O . Considérons deux cas. Si O est une sphère caractérisée par un rayon $r < 1/2$, alors la longueur moyenne d'un rayon issu de \mathcal{R}_R s'écrit de la manière suivante :

$$\bar{\nu}_{\Sigma \cup \partial R} = \begin{cases} \frac{1}{3 + 4\pi r^2} & \text{si la sphère est non opaque} \\ \frac{3 - 4\pi r^3}{3(3 + 2\pi r^2)} & \text{sinon} \end{cases} .$$

Si O est un cube caractérisé par une longueur $a < 1$ alors la longueur moyenne d'un rayon issu de \mathcal{R}_R s'écrit de la manière suivante :

$$\bar{\nu}_{\Sigma \cup \partial R} = \begin{cases} \frac{1}{3(1+2a^2)} & \text{si le cube est non opaque} \\ \frac{1-a^3}{3(1+a^2)} & \text{sinon} \end{cases} .$$

Les différentes longueurs moyennes obtenues sont illustrées sur les graphes suivants.



□

4.4.5 Nombre de régions traversées par un rayon

Soit R un voxel. Soit $\mathcal{S} = \{R_k\}_{k=1}^n$ une subdivision uniforme de R . Soient O^1, \dots, O^N des objets et Σ l'ensemble des surfaces optiques (définition 5 page 96) associées à ces objets. Soit $R_{x,w}$ un rayon traversant R . À ce rayon est associé un segment, noté $A_{x,w}$, de longueur $\nu_{\Sigma \cup \partial R}(x, w)$. Le nombre de régions de \mathcal{S} traversées par $R_{x,w}$ est le nombre de régions de \mathcal{S} traversées par le segment $A_{x,w}$. Il est donné par la proposition 16. L'expression 4.3 est très intéressante puisqu'elle s'affranchit de la position initiale du segment (pour l'évaluation du nombre moyen de régions traversées, car il subsiste la position initiale pour l'expression de la longueur du segment) :

$$\bar{m}_{A_{x,w}} = 1 + \nu_{\Sigma \cup \partial R}(x, w) \sum_{i=0}^2 \frac{|w_i|}{\delta_i} .$$

Par définition du nombre moyen de régions traversées par un rayon issu de \mathcal{R}_R , on a :

$$\bar{m} = \frac{1}{\text{mes}(\mathcal{R}_R)} \int_{\mathcal{R}_R} m(x, w) d\mu(x) d\sigma(w) ,$$

où $m(x, w)$ est le nombre moyen de régions de \mathcal{S} traversées par le rayon $R_{x,w}$. Cependant, ne connaissant pas exactement $m(x, w)$, on approche la quantité \bar{m} par la quantité \bar{m}' définie de la manière suivante :

$$\bar{m}' = \frac{1}{\text{mes}(\mathcal{R}_R)} \int_{\mathcal{R}_R} \bar{m}_{A_{x,w}} d\mu(x) d\sigma(w) ,$$

La proposition suivante donne une expression du nombre \bar{m}' .

Proposition 27

Soit R un voxel. Soit \mathcal{S} une subdivision uniforme de R . Soient O^1, \dots, O^N des objets optiquement cohérents. Le nombre moyen de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R est donné par :

$$\bar{m}' = 1 + \frac{1}{2} \bar{\nu}_{\Sigma \cup \partial R} \sum_{i=0}^2 \frac{1}{\delta_i} , \quad (4.38)$$

où $\bar{\nu}_{\Sigma \cup \partial R}$ est la longueur moyenne d'un rayon issu de \mathcal{R}_R .

□

Preuve On part de l'expression du nombre moyen de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R :

$$\bar{m}' = \frac{1}{\text{mes}(\mathcal{R}_R)} \int_{\mathcal{R}_R} \bar{m}_{A_{x,w}} d\mu(x) d\sigma(w) .$$

L'intégrale dans l'expression précédente s'écrit de la manière suivante :

$$\begin{aligned} \int_{\mathcal{R}_R} \bar{m}_{A_{x,w}} d\mu(x) d\sigma(w) &= \int_{\mathcal{R}_R} \left(1 + \nu_{\Sigma \cup \partial R}(x, w) \sum_{i=0}^2 \frac{|w_i|}{\delta_i} \right) d\mu(x) d\sigma(w) \\ &= \text{mes}(\mathcal{R}_R) + \sum_{i=0}^2 \left(\frac{1}{\delta_i} \int_{\mathcal{R}_R} \nu_{\Sigma \cup \partial R}(x, w) |w_i| d\mu(x) d\sigma(w) \right) . \end{aligned}$$

Pour calculer l'intégrale de cette expression, on décompose par rapport à chaque variable, en utilisant la fonction $\ell_{\Sigma \cup \partial R}$ qui est le prolongement sur $(\Sigma \cup \partial R) \times \mathcal{S}^2$ de la fonction $\nu_{\Sigma \cup \partial R}$:

$$\begin{aligned} \int_{\mathcal{R}_R} \nu_{\Sigma \cup \partial R}(x, w) |w_i| d\mu(x) d\sigma(w) &= \int_{(\Sigma \cup \partial R) \times \mathcal{S}^2} \ell_{\Sigma \cup \partial R}(x, w) |w_i| d\mu(x) d\sigma(w) \\ &= \int_{\mathcal{S}^2} \left(\int_{\Sigma \cup \partial R} \ell_{\Sigma \cup \partial R}(x, w) d\mu(x) \right) |w_i| d\sigma(w) . \end{aligned}$$

Or, comme on l'a écrit au cours de la démonstration de la proposition 26, l'intégrale de la longueur d'un rayon sur $\Sigma \cup \partial R$ s'écrit de la manière suivante :

$$\int_{\Sigma \cup \partial R} \ell_{\Sigma \cup \partial R}(x, w) d\mu(x) = \text{vol}(R) - \sum_{i=1}^N v_i ,$$

où les $\{v_i\}_i$ sont donnés par l'expression 4.36. Cette expression est indépendante de w , on peut donc écrire :

$$\begin{aligned} \int_{\mathcal{R}_R} \nu_{\Sigma \cup \partial R}(x, w) |w_i| \delta_i d\mu(x) d\sigma(w) &= \left(\int_{\mathcal{S}^2} |w_i| d\sigma(w) \right) \left(\text{vol}(R) - \sum_{i=1}^N v_i \right) \\ &= 2\pi \left(\text{vol}(R) - \sum_{i=1}^N v_i \right) . \end{aligned}$$

Il ne reste plus qu'à injecter cette expression dans l'expression de la complexité moyenne en temps :

$$\bar{m}' = \frac{1}{\text{mes}(\mathcal{R}_R)} \left(\text{mes}(\mathcal{R}_R) + \sum_{i=0}^2 \frac{2\pi}{\delta_i} \left(\text{vol}(R) - \sum_{i=1}^N v_i \right) \right) ,$$

soit finalement en regroupant les termes et en utilisant l'expression de la mesure de l'ensemble \mathcal{R}_R donnée par 4.31, on obtient :

$$\bar{m}' = 1 + \frac{\text{vol}(R) - \sum_{i=1}^N v_i}{\text{aire}(\partial R) + \sum_{i=1}^N a_i} \sum_{i=0}^2 \frac{1}{\delta_i} ,$$

où les $\{a_i\}_i$ sont donnés par l'expression 4.37. Le second terme de l'expression précédente n'est autre que la moitié de la longueur moyenne donnée par l'expression 4.35 de la proposition 26. D'où l'expression proposée pour le nombre moyen de régions traversées.

■

Exemple 12

Reprenons l'exemple 11. Soit R un cube de côté 1. Soit une scène géométrique incluse dans ce cube et contenant un seul objet O . Soit une subdivision uniforme \mathcal{S} de R telle que $\delta_i = \delta$ dans chaque direction. Autrement dit, le nombre de subdivisions dans chaque direction est :

$$n_i = n^{1/3} .$$

Considérons deux cas. Si O est une sphère caractérisée par un rayon $r < 1/2$, alors le nombre moyen de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R s'écrit de la manière suivante :

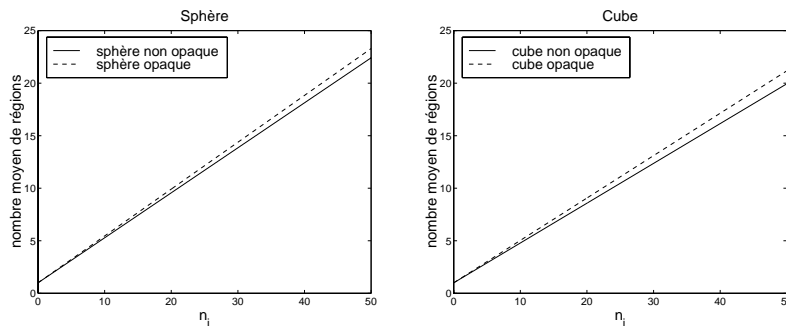
$$\overline{m'} = \begin{cases} 1 + \frac{3}{2(3 + 4\pi r^2)} n_i & \text{si la sphère est non opaque} \\ 1 + \frac{3 - 4\pi r^3}{2(3 + 2\pi r^2)} n_i & \text{sinon} \end{cases} .$$

Si O un cube caractérisé par une longueur $a < 1$ alors le nombre moyen de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R s'écrit de la manière suivante :

$$\overline{m'} = \begin{cases} 1 + \frac{1}{2(1 + 2a^2)} n_i & \text{si le cube est non opaque} \\ 1 + \frac{1 - a^3}{2(1 + a^2)} n_i & \text{sinon} \end{cases} .$$

Les différents nombres de régions traversées, en fonction du nombre de subdivisions n_i , sont illustrés sur les graphes suivants. Pour ce faire, nous avons fixé les valeurs des paramètres de la sphère et du cube :

$$\begin{cases} r = 0.2 \\ a = 0.4 \end{cases} .$$



L'influence de la propriété d'opacité des objets sur le nombre moyen de régions traversées est très faible pour les paramètres considérés.

□

4.4.6 Nombre d'objets rencontrés par un rayon

Soit $R_{x,w}$ un rayon. Soit R un voxel. Soit $\mathcal{S} = \{R_k\}_{k=1}^n$ une subdivision de R . Soient O^1, \dots, O^N des objets. Le nombre $N(x, w)$ introduit dans le corollaire 8 représente le nombre d'objets rencontrés par $R_{x,w}$ lors du parcours de \mathcal{S} . Le nombre est dépendant de la géométrie de la scène et de la subdivision considérées. Donner une expression exacte de $N(x, w)$ est un problème ouvert dont les notions nécessaires à sa formulation font appels à des techniques de géométrie discrète, éventuellement de géométrie intégrale. C'est pourquoi dans ce paragraphe on propose de donner une estimation du nombre d'objets rencontrés par un rayon.

La première propriété que l'on peut donner est la suivante :

$$0 \leq N(x, w) \leq N .$$

Cette propriété traduit le fait que le nombre d'objets rencontrés est nécessairement positif et inférieur au nombre d'objets considérés (lors du calcul d'intersection, un même objet n'est considéré qu'une seule fois, même s'il est rencontré dans plusieurs régions). On ne peut en rester à un tel encadrement, la borne supérieure correspondant à un calcul d'intersection avec tous les objets O^1, \dots, O^N , c'est-à-dire qu'on ne prend pas en compte la subdivision. Pour affiner l'estimation de $N(x, w)$ deux solutions sont envisageables. La première consiste à définir un encadrement plus fin. Soit N_{\max} le nombre maximum d'objets dans une région de \mathcal{S} :

$$N_{\max} = \max \{ \text{card}(L_{R_k}), k = 1, \dots, n \} ,$$

où L_{R_k} est donné par la définition 6 page 96. D'où un encadrement de $N(x, w)$:

$$0 \leq N(x, w) \leq m(x, w) N_{\max} ,$$

où $m(x, w)$ est le nombre de régions traversées par $R_{x,w}$ dans \mathcal{S} . Le nombre d'objets rencontrés au maximum est bien le nombre de régions parcourues multiplié par le nombre d'objets maximum liés à une région. Lorsque la subdivision s'affine, c'est-à-dire lorsque n augmente, $m(x, w)$ augmente et N_{\max} diminue. La borne supérieure de l'encadrement précédent n'est pas satisfaisante. En effet, $m(x, w) N_{\max}$ n'est pas nécessairement inférieur à N .

La seconde solution consiste à définir une approximation de $N(x, w)$ plutôt qu'un encadrement. Soit N_{moy} le nombre moyen d'objets dans une région de \mathcal{S} :

$$N_{\text{moy}} = \frac{1}{n} \sum_{k=1}^n \text{card}(L_{R_k}) = \frac{1}{n} \sum_{i=1}^N \text{card}(L_{\Upsilon(O^i)}) ,$$

ce que finalement l'on peut écrire :

$$N_{\text{moy}} = \frac{1}{n} \sum_{i=1}^N m_{\Upsilon(O^i)} . \quad (4.39)$$

Pour évaluer ce nombre, on utilise les résultats de la section 4.2. On a alors une expression approchée du nombre moyen d'objets par région :

$$\bar{N}_{\text{moy}} = \frac{1}{n} \sum_{i=1}^N \bar{m}_{\Upsilon(O^i)} . \quad (4.40)$$

On peut alors approcher le nombre $N(x, w)$ d'objets rencontrés par $R_{x,w}$ par la quantité suivante :

$$m(x, w) \bar{N}_{\text{moy}} .$$

L'expression moyenne de cette quantité, au sens de la définition 1 page 36, par rapport à l'ensemble des rayons issus de \mathcal{R}_R s'écrit :

$$\frac{1}{\text{mes}(\mathcal{R}_R)} \int_{\mathcal{R}_R} m(x, w) \overline{N}_{\text{moy}} d\mu(x) d\sigma(w) = \overline{m} \overline{N}_{\text{moy}} ,$$

et, d'après le paragraphe 4.4.5, peut être approchée par la quantité suivante :

$$\overline{N}' = \overline{m}' \overline{N}_{\text{moy}} . \quad (4.41)$$

\overline{N}' est une valeur approchée de \overline{N} . Cette estimation donne une approximation en fonction des paramètres géométriques considérés.

4.4.7 Estimation

L'évaluation des différents paramètres introduits lors de la caractérisation de la complexité en temps permet d'en donner une estimation. Pour ce faire, on se place dans le contexte du corollaire 8 et de la proposition 25, où l'on considère que les complexités en temps de tous les objets sont identiques. La proposition 25 donne la complexité moyenne en temps pour un ensemble de rayons issus de \mathcal{R}_R , où \mathcal{R}_R est donné au paragraphe 4.4.3. Pour utiliser les estimations des paramètres de l'expression 4.29, il est nécessaire d'introduire la complexité moyenne en temps approchée $\overline{T}'_{\mathcal{S}, O^1, \dots, O^N}$. Soit R un parallélépipède. Soit $\{R_k\}_{k=1}^n$ une subdivision uniforme de R . La complexité moyenne en temps $\overline{T}'_{\mathcal{S}, O^1, \dots, O^N}$ est donnée par l'expression suivante :

$$\boxed{\overline{T}'_{\mathcal{S}, O^1, \dots, O^N} = T_{\text{psv}} + \overline{m}' (T_v + \overline{N}_{\text{moy}} T_O)} , \quad (4.42)$$

où les différents paramètres sont donnés dans les paragraphes précédents. Les complexités en temps T_{psv} et T_v sont propres aux algorithmes 10, 11 et 12. La complexité en temps T_O est propre aux objets pris en compte. La quantité \overline{m}' représente le nombre moyen approché de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R , elle est donnée par la proposition 27 et s'écrit :

$$\overline{m}' = 1 + \frac{1}{2} \overline{\nu}_{\Sigma \cup \partial R} \sum_{i=0}^2 \frac{1}{\delta_i} ,$$

où $\overline{\nu}_{\Sigma \cup \partial R}$ est la longueur moyenne d'un rayon issu de \mathcal{R}_R , donné par la proposition 26 et s'écrit :

$$\overline{\nu}_{\Sigma \cup \partial R} = \frac{2 \left(\text{vol}(R) - \sum_{i=1}^N v_i \right)}{\text{aire}(\partial R) + \sum_{i=1}^N a_i} ,$$

où v_i représente le volume pour les objets (volumiques) opaques, il est donné par l'expression 4.36 :

$$v_i = \begin{cases} \text{vol}(O^i) & \text{si } O^i \text{ est un objet volumique opaque} \\ 0 & \text{sinon} \end{cases} ,$$

et a_i caractérise l'aire des surfaces optiques, il est donné par l'expression 4.37 :

$$a_i = \begin{cases} \text{aire}(\partial O^i) & \text{si } O^i \text{ est un objet volumique opaque} \\ 2 \text{aire}(\Sigma(O^i)) & \text{sinon} \end{cases} ,$$

où $\Sigma(O^i)$ est la surface optique associée à l'objet O^i . \bar{N}_{moy} représente le nombre moyen approché d'objets par régions de \mathcal{S} , il est donné par l'expression 4.40 :

$$\bar{N}_{\text{moy}} = \frac{1}{n} \sum_{i=1}^N \bar{m}_{\Upsilon(O^i)},$$

où $\bar{m}_{\Upsilon(O^i)}$ est le nombre de régions de \mathcal{S} traversées par l'objet O^i , au sens de la localisation envisagée pour cet objet. On se reportera à la section 4.3.2 pour différentes expressions de $\bar{m}_{\Upsilon(O^i)}$.

Exemple 13

Soit R un cube de côté 1. Soit une scène géométrique incluse dans ce cube et composée de N sphères O^1, \dots, O^N optiquement cohérentes deux à deux, non opaques, de rayon r , tel que :

$$r < \frac{1}{2N^{1/3}}.$$

La longueur moyenne d'un rayon issu de \mathcal{R}_R s'écrit :

$$\bar{v}_{\Sigma \cup \partial R} = \frac{1}{3 + 4N\pi r^2}.$$

Soit \mathcal{S} une subdivision uniforme de R telle que $\delta_i = \delta$ dans chaque direction. Autrement dit, le nombre de subdivisions dans chaque direction est :

$$n_i = n^{1/3}.$$

Le nombre moyen approché de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R s'écrit :

$$\bar{m}' = 1 + \frac{3}{2} \bar{v}_{\Sigma \cup \partial R} n_i.$$

Supposons que chaque sphère est localisée à l'aide d'une boîte englobante associée, un cube de côté $2r$. La localisation par rapport à l'ensemble de objets O^1, \dots, O^N donne le nombre (exact) de régions de \mathcal{S} qui ont une intersection avec l'ensemble des objets considérés :

$$\bar{m}_{\Upsilon(O^i)} = m_{B(O^i)} = (2r n_i + 1)^3.$$

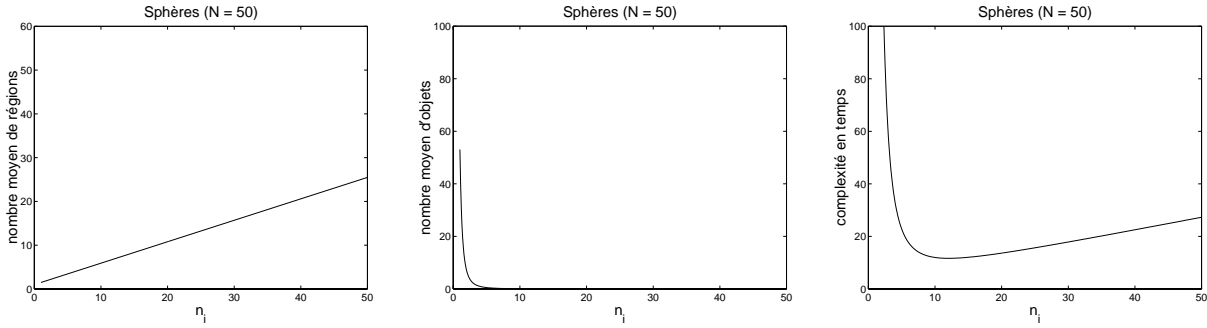
D'où l'expression du nombre moyen (exact) d'objets par région de \mathcal{S} :

$$\bar{N}_{\text{moy}} = N n_i^{-3} (2r n_i + 1)^3.$$

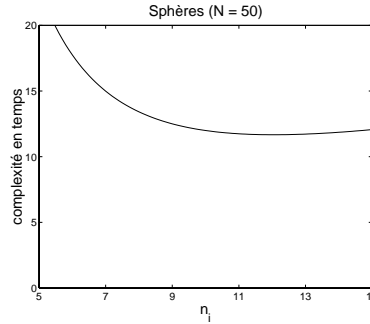
Pour illustrer ces différentes quantités, et surtout la complexité moyenne en temps approchée, donnée par l'expression 4.42, considérons les valeurs suivantes pour les paramètres intervenant dans 4.42 :

$$\left\{ \begin{array}{l} T_{\text{psv}} = 1 \\ T_v = 1 \\ T_O = 10 \\ N = 50 \\ r = 0.01 \end{array} \right. .$$

Les graphes suivants illustrent le nombre moyen de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R en fonction du nombre de subdivisions n_i , le nombre moyen d'objets par régions de \mathcal{S} et la complexité moyenne en temps approchée.



La complexité moyenne en temps approchée a un minimum pour $n_i = 12$.



□

Exemple 14

Soit R un cube de côté 1. Soit une scène géométrique incluse dans ce cube et composée de N triangles O^1, \dots, O^N optiquement cohérents deux à deux, de formes identiques, c'est dire de périmètre et d'aire identiques. L'ensemble de ces triangles décrit un objet surfacique. Ainsi, la longueur moyenne d'un rayon issu de \mathcal{R}_R s'écrit :

$$\bar{\nu}_{\Sigma \cup \partial R} = \frac{1}{3 + N \text{aire}(T)},$$

où T est un triangle. Soit \mathcal{S} une subdivision uniforme de R telle que $\delta_i = \delta$ dans chaque direction. Autrement dit, le nombre de subdivisions dans chaque direction est :

$$n_i = \frac{1}{\delta_i} = n^{1/3}.$$

Le nombre moyen approché de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R s'écrit :

$$\bar{m}' = 1 + \frac{3}{2} \bar{\nu}_{\Sigma \cup \partial R} n_i.$$

Supposons que chaque triangle est localisé par rapport à sa surface. Le nombre de régions de \mathcal{S} traversées par un triangle est donné par l'expression 4.11 :

$$\bar{m}_T = 1 + \frac{1}{2\delta} \sum_{j=1}^3 \sum_{i=0}^2 |w_i^j| \ell_j + \frac{1}{\delta^2} \sum_{i=0}^2 |u_i| \text{aire}(T).$$

Pour simplifier les écritures, considérons une moyenne de cette quantité par rapport aux directions considérées :

$$\langle \bar{m}_T \rangle = 1 + \frac{3}{4} \text{perim}(T) n_i + \frac{3}{2} \text{aire}(T) n_i^2.$$

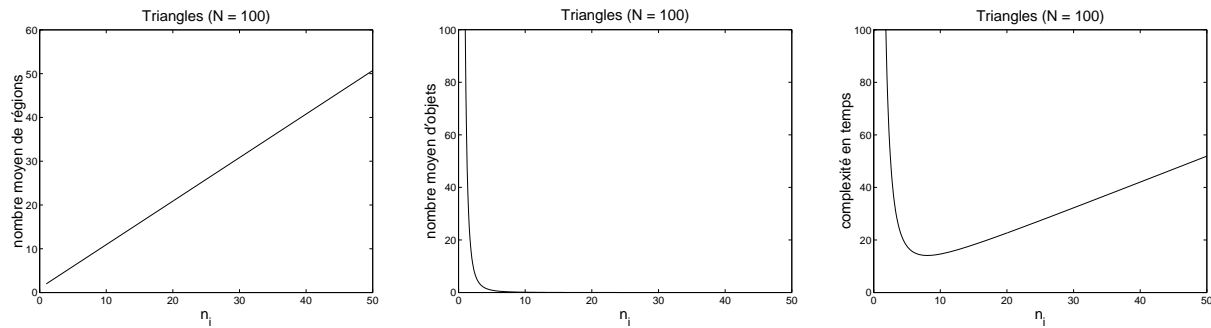
D'où l'expression du nombre moyen approché d'objets par région de \mathcal{S} , moyennée par rapport à l'ensemble des directions :

$$\langle \bar{N}_{\text{moy}} \rangle = N \left(n_i^{-3} + \frac{3}{4} \text{perim}(T) n_i^{-2} + \frac{3}{2} \text{aire}(T) n_i^{-1} \right).$$

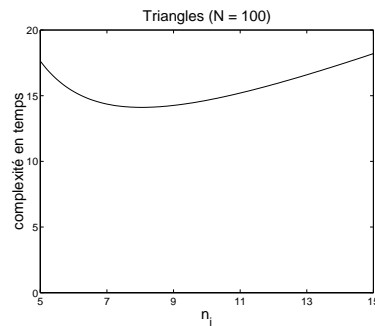
Pour illustrer ces différentes quantités, et surtout la complexité moyenne en temps approchée, donnée par l'expression 4.42, considérons les valeurs suivantes pour les paramètres intervenant dans 4.42 :

$$\begin{cases} T_{\text{psv}} & = 1 \\ T_v & = 1 \\ T_O & = 2 \\ N & = 100 \\ \text{perim}(T) & = 0.03 \\ \text{aire}(T) & = 8.66 \cdot 10^{-5} \end{cases}.$$

Les graphes suivants illustrent le nombre moyen de régions de \mathcal{S} traversées par un rayon issu de \mathcal{R}_R en fonction du nombre de subdivisions n_i , le nombre moyen d'objets par régions de \mathcal{S} et la complexité moyenne en temps approchée.



La complexité moyenne en temps approchée a un minimum pour $n_i = 8$.



□

4.5 Une nouvelle méthode de choix des paramètres de subdivision

On rappelle que l'objectif d'une méthode de subdivision est de diminuer le coût en temps des algorithmes de recherche de la première intersection entre des rayons et des objets. Les paramètres

d'une subdivision uniforme \mathcal{S} d'un parallépipède R sont les nombres de subdivisions dans chaque direction i :

$$n_i = \frac{\Delta_i}{\delta_i} ,$$

où Δ_i est la dimension dans la direction i de R (figure 4.19).

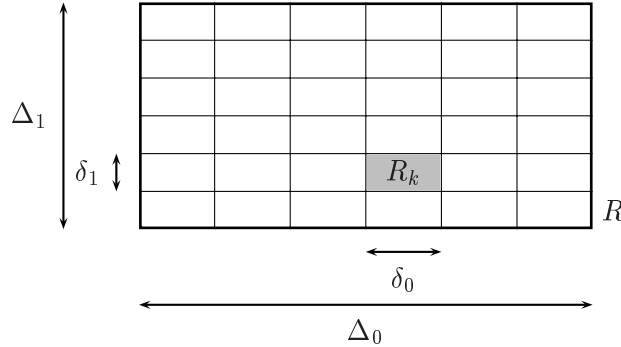


Fig. 4.19 – Rappel des notations pour une subdivision uniforme.

Le meilleur choix pour (n_0, n_1, n_2) est donc celui qui minimise la complexité moyenne en temps, caractérisée dans la section 4.4. Pour être conforme aux développements qui viennent d'être présentés, on considère l'expression 4.42 de la complexité moyenne en temps approchée :

$$\overline{T}'_{\mathcal{S}, O^1, \dots, O^N} = T_{\text{psv}} + \overline{m}' (T_v + \overline{N}_{\text{moy}} T_O) .$$

Cependant, il ne faut pas perdre de vue que les simulations ont lieu sur des machines dont les capacités mémoire sont limitées. Ainsi, on doit veiller à ce que le triplet (n_0, n_1, n_2) choisi donne une complexité en mémoire, caractérisée dans la section 4.3, qui soit acceptable. De même que pour la complexité moyenne en temps, on considère l'expression 4.26 de la complexité en mémoire approchée :

$$M'_{\mathcal{S}, O^1, \dots, O^N} = \sum_{i=1}^N (M_{O^i} + \overline{m}_{\Upsilon(O^i)} M_{v_o}) .$$

Soient O^1, \dots, O^N des objets optiquement cohérents et R un parallépipède englobant ces objets. Pour déterminer les paramètres d'une subdivision uniforme \mathcal{S} de R , on propose de calculer a priori la complexité en mémoire et la complexité moyenne en temps. Pour ce faire, on commence par déterminer le type de localisation envisagée pour chaque objet, c'est-à-dire que l'on explicite $\Upsilon(O^1), \dots, \Upsilon(O^N)$, où l'on rappelle que $\Upsilon(O^i)$ est l'objet envisagé pour la localisation de l'objet O^i . Ensuite, on évalue la complexité en mémoire et la complexité moyenne en temps pour différentes valeurs du triplet $(n_0, n_1, n_2) \in (\mathbb{N}^*)^3$, et on ne conserve que celui qui minimise la complexité moyenne en temps, tout en restant acceptable pour la complexité en mémoire.

L'indice de performance globale, introduit au paragraphe 3.3.3, est fonction des paramètres n_0, n_1 et n_2 . Il est donné théoriquement par :

$$\rho'(n_0, n_1, n_2) = \frac{M'_{\mathcal{S}, O^1, \dots, O^N} \overline{T}'_{\mathcal{S}, O^1, \dots, O^N}}{\sum_{i=1}^N M_{O^i} N T_O} .$$

$\rho'(n_0, n_1, n_2)$ est une valeur approchée puisque $M'_{\mathcal{S}, O^1, \dots, O^N}$ et $\overline{T}'_{\mathcal{S}, O^1, \dots, O^N}$ sont des quantités approchées. Pour calculer les complexités qui nous intéressent, ainsi que l'indice de performance globale, il est nécessaire d'évaluer les quantités suivantes pour un triplet (n_0, n_1, n_2) donné :

- $\overline{m}_{\Upsilon(O^i)}$, le nombre moyen de régions de \mathcal{S} traversées par $\Upsilon(O^i)$ (section 4.2),
- $\overline{\nu}_{\Sigma \cup \partial R}$, la longueur moyenne d'un rayon (paragraphe 4.4.4),
- \overline{m}' , le nombre moyen de régions de \mathcal{S} traversées par un rayon (paragraphe 4.4.5),
- $\overline{N}_{\text{moy}}$, le nombre moyen d'objets rencontrés par un rayon (paragraphe 4.4.6).

La recherche des bons paramètres, au sens du critère consistant à minimiser la complexité moyenne en temps, tout en s'assurant d'une complexité en mémoire acceptable, revient à chercher les plus grands nombres de subdivisions pour lesquels l'indice $\rho'(n_0, n_1, n_2)$ est inférieur à un certain seuil. Le dénominateur de $\rho'(n_0, n_1, n_2)$ est constant, il correspond au produit de la complexité moyenne en temps par la complexité en mémoire pour la méthode naïve (algorithme 6 page 90). Seul le numérateur varie. On propose d'évaluer $\rho'(n_0, n_1, n_2)$ pour différentes valeurs du triplet (n_0, n_1, n_2) et de retenir celui qui minimise cette quantité. Les expérimentations menées au chapitre 3 montrent qu'il n'est pas efficace de trop subdiviser. On pourra donc se contenter de valeurs pour (n_0, n_1, n_2) dans $I_0 \times I_1 \times I_2 \subset (\mathbb{N}^*)^3$ où I_0, I_1 et I_2 sont des sous-ensembles de \mathbb{N}^* de petites tailles (par exemple $[0, 10]$). Ce qui fait au plus 10^3 évaluations de la complexité moyenne en temps $\overline{T}'_{\mathcal{S}, O^1, \dots, O^N}$ et de la complexité en mémoire $M'_{\mathcal{S}, O^1, \dots, O^N}$.

L'algorithme suivant permet de choisir les bons paramètres de subdivision, au sens du critère considéré, en fonction des ensembles I_0, I_1 et I_2 et d'une limite supérieure M_{\max} pour la complexité en mémoire. Il minimise la complexité moyenne en temps par rapport aux variables n_0, n_1 et n_2 sous la contrainte d'une complexité en mémoire maximale.

Algorithme 16: Recherche des paramètres de subdivision optimaux.

PARSUB(I_0, I_1, I_2, M_{\max})

- (1) **for each** $(n_0, n_1, n_2) \in I_0 \times I_1 \times I_2$
- (2) $\overline{T}_{\min} \leftarrow +\infty$
- (3) **for** $i = 1$ **to** N
- (4) évaluer $\overline{m}_{\Upsilon(O^i)}$
- (5) évaluer $\overline{T}'_{\mathcal{S}, O^1, \dots, O^N}$
- (6) **if** $\overline{T}'_{\mathcal{S}, O^1, \dots, O^N} < \overline{T}_{\min}$
- (7) évaluer $M'_{\mathcal{S}, O^1, \dots, O^N}$
- (8) **if** $M'_{\mathcal{S}, O^1, \dots, O^N} < M_{\max}$
- (9) $(n_{\min,0}, n_{\min,1}, n_{\min,2}) \leftarrow (n_0, n_1, n_2)$
- (10) **return** $(n_{\min,0}, n_{\min,1}, n_{\min,2})$

La complexité en temps de l'algorithme 16 est due au choix des ensembles I_0, I_1 et I_2 et aux évaluations de la complexité moyenne en temps et de la complexité en mémoire. Il choisir des ensembles I_0, I_1 et I_2 au plus juste, c'est-à-dire d'avoir une borne inférieure suffisamment grande, il n'est pas nécessaire de toujours prendre 1, et une borne supérieure pas trop grande, que l'on pourra limiter en fonction du nombre d'objets N à considérer. Il faut adopter une stratégie efficace pour le parcours de $I_0 \times I_1 \times I_2$. On commencera par les plus petites valeurs du triplet (n_0, n_1, n_2) .

4.6 Mise en place d'une hiérarchie de subdivisions uniformes

L'algorithme 16 donne un moyen d'évaluer les paramètres optimaux pour une subdivision uniforme. Dans le cas d'une hiérarchie de subdivisions uniformes, le même algorithme peut-être utilisé pour chaque nœud de l'arbre des subdivisions. Cependant, il faut tenter de minimiser la complexité en temps de l'arbre des subdivisions, ce qui ne veut pas nécessairement dire minimiser la complexité en temps pour chaque nœud de l'arbre. De plus, il faut prendre en compte la complexité en mémoire.

Ce constat amène à se poser naturellement la question de l'évaluation a priori de la complexité moyenne en temps et de la complexité en mémoire pour une structure hiérarchique. Les évaluations présentées dans ce chapitre concernant les subdivisions uniformes. Elles nécessitent la connaissance des objets (propriétés géométriques et optiques). Dans le contexte d'une hiérarchie ces évaluations sont locales. À chaque nœud de l'arbre d'une hiérarchie de subdivisions uniformes on est donc en mesure de connaître a priori la complexité moyenne en temps et la complexité en mémoire associées, à condition de connaître les objets à considérer. Pour ce faire, il faut construire au fur et à mesure la structure de données associée de manière à déterminer la répartition des objets au niveau des différents nœuds. Ainsi, ce n'est qu'une fois toute la structure de données construite que l'on peut connaître les complexités. Ce n'est donc pas une évaluation a priori.

La solution pour déterminer la hiérarchie de subdivisions uniformes qui consiste à minimiser la complexité moyenne en temps, en évaluant les différentes configurations arborescentes, tout en obtenant une complexité en mémoire acceptable n'est donc pas envisageable. En effet, le nombre d'évaluations de la complexité moyenne en temps et de la complexité en mémoire est très grand, et leur coût serait trop important pour un pré-calcul à la simulation.

On peut cependant utiliser des éléments expérimentaux et intuitifs pour guider la recherche d'une hiérarchie de subdivisions uniformes donnant une complexité moyenne en temps la plus faible possible. Expérimentalement, on s'aperçoit qu'il n'est pas nécessaire de trop subdiviser au début. Ainsi, on peut utiliser l'algorithme 16 en choisissant bien les ensembles I_0 , I_1 et I_2 . Des expérimentations seraient nécessaires pour orienter ces choix. Néanmoins, on peut déjà, à la vue des résultats des tests du chapitre 3, avoir une idée des ensembles à choisir et de la profondeur maximale de la hiérarchie.

Cette méthode a l'avantage de construire la structure de données une seule fois. La complexité en temps de recherche de la meilleure hiérarchie de subdivisions uniformes est ainsi tout à fait acceptable pour un pré-calcul.

4.7 Conclusion

Ce chapitre donne des expressions pour la complexité moyenne en temps et la complexité en mémoire des méthodes liées à la localisation spatiale par subdivision uniforme. Le développement théorique présenté ici s'applique à des objets linéiques (segment, ligne brisée) et à des objets surfaciques ou volumiques (polygone plan, surface affine par morceaux). Les boîtes englobantes, souvent utilisées pour la localisation, sont aussi considérées. Une stratégie pour la recherche de la meilleure subdivision uniforme, au sens d'une complexité moyenne en temps minimale et d'une complexité en mémoire acceptable, est présentée. Elle est étendue à une hiérarchie de subdivisions uniformes.

Ces résultats constituent une approche nouvelle pour la recherche d'un paramétrage optimal des méthodes d'accélération des calculs par localisation spatiale par subdivision uniforme. Les différents paramètres exhibés (nombre de régions traversées par un objet, longueur moyenne d'un rayon, nombre d'objets rencontrés par un rayon) peuvent être utilisés pour d'autres méthodes d'accélération.

Perspectives

Le travail accompli au cours de cette thèse laisse envisager, d'une part, des applications immédiates des algorithmes proposés ainsi que des résultats théoriques et, d'autre part, des perspectives de recherches intéressantes.

L'étude théorique menée au chapitre 4 peut être adaptée sans difficulté à un code de calcul utilisant une localisation par des subdivisions uniformes. De nombreux travaux dans le domaine de l'image de synthèse sont concernés (méthodes globales et mixtes décrites dans la section 3.1.3). Concernant le logiciel Speos, une réécriture d'une partie du code serait nécessaire pour stabiliser le logiciel, la mise en œuvre des critères de choix automatiques pour les paramètres de localisation peuvent alors être considérés sans difficulté dans les outils développés au cours du projet.

L'estimation du nombre de régions d'une subdivision uniforme traversées par un objet (section 4.2) a été donnée pour la ligne brisée, la surface affine par morceaux et la boîte englobante). Les propositions 17 page 147 (ligne brisée) et 20 page 158 (surface affine par morceaux) peuvent être généralisées pour des objets continus (courbe et surface dérivables). De manière analogue à la proposition 22 page 162 (boîte englobante), une estimation du nombre moyen de régions traversées par un polyèdre est envisageable, ainsi qu'une généralisation pour des objets volumiques dont la frontière est une surface fermée dérivable.

La géométrie intégrale (Santoló [72] [73], DeLin [31]) est un domaine des mathématiques qui apporte des idées et des éléments de réponse très intéressants par rapport aux problèmes posés dans le chapitre 4. Buffon a posé et résolu en 1777 le problème suivant : étant donné un ensemble de droites parallèles dans \mathbb{R}^2 séparées d'une distance fixe, quelle est la probabilité qu'un segment de longueur donnée ait une intersection avec au moins une droite. Ce problème a été généralisé à des courbes fermées convexes, puis étendu sous le nom de problème de Laplace à deux ensembles orthogonaux de droites parallèles séparées par des distances fixes. On obtient ainsi un ensemble de régions congruentes engendré par un rectangle, c'est-à-dire une subdivision uniforme rectangulaire de \mathbb{R}^2 . L'utilisation du formalisme attractif de ce domaine ainsi que des résultats existants laisse envisager des solutions très intéressantes pour les problèmes qui nous préoccupent dans \mathbb{R}^3 .

Les algorithmes de calcul envisagés pour les simulations numériques qui nous concernent sont tributaires de la géométrie des éléments mis en jeu. Le constat principal faisant suite à l'étude des complexités en temps de ces algorithmes est que les calculs d'intersection entre des rayons et des objets sont très coûteux. Il serait très intéressant de pouvoir s'en affranchir. Au cours de la thèse et en dehors des contraintes industrielles, une étude a été engagée dans ce sens, et les premières expériences réalisées sont encourageantes. Ainsi, la notion d'équivalent optique a été introduite. Elle consiste à remplacer une partie d'un système optique par un objet (une boîte par exemple), appelé équivalent optique, et à définir sur les contours de cet objet des fonctions qui décrivent entièrement les caractéristiques géométriques et physiques de la partie du système remplacée. L'équivalent optique fonctionne comme un système prenant en entrée un signal (par exemple la luminance entrante) et rendant en sortie un autre signal (par exemple luminance sortante). Ainsi, les calculs d'intersections ne sont pas effectués en

cours de simulation. Bien évidemment, le compromis entre la complexité en mémoire et la complexité en temps reste présent dans ces premières expériences, et, pour l'instant, cette méthode se heurte au problème de la gestion d'une très grande quantité d'information. Comme dans tous les problèmes faisant intervenir des fonctions, le choix des bases est crucial, et c'est dans cette direction que doit s'orienter la majeure partie du travail de recherche sur ce sujet.

Bibliographie

- [1] Bruno Arnaldi, Thierry Priol, et Kadi Bouatouch.
A new space method for ray tracing CSG modelled scenes.
The Visual Computer, 3(2) :98–108, August 1987.
- [2] James Arvo.
Transfer Equations in Global Illumination.
In *SIGGRAPH '93 Course Notes - Global Illumination*, chapter 1, pages 1–28. ACM Press, August 1993.
- [3] James Arvo.
Analytic Methods for Simulated Light Transport.
PhD thesis, Yale University, December 1995.
- [4] James Arvo et David Kirk.
Particle Transport and Image Synthesis.
Computer Graphics (SIGGRAPH '90 Proceedings), 24(4) :63–66, August 1990.
- [5] Alan H. Barr.
Ray Tracing Deformed Surfaces.
Computer Graphics (SIGGRAPH '86 Proceedings), 20(4) :287–296, August 1986.
- [6] Luc Biard.
Méthode algorithmique d'implicitisation et d'inversion. Application au lancer de rayons.
Thèse en mathématiques appliquées, Laboratoire LMC-IMAG, Université Joseph Fourier, Grenoble, Novembre 1990.
- [7] James F. Blinn.
A Generalization of Algebraic Surface Drawing.
ACM Transactions on Graphics, 1(3) :235–256, July 1982.
- [8] Daniel Bouche et Frédéric Molinet.
Méthodes asymptotiques en électromagnétisme, volume 16 of *Mathématiques et Applications*.
Springer-Verlag, Berlin, Heidelberg, New-York, 1994.
- [9] Christian Bouville.
Bounding Ellipsoids for Ray-Fractal Intersection.
Computer Graphics (SIGGRAPH '85 Proceedings), 19(3) :45–52, July 1985.
Held in San Francisco, California.
- [10] Jack E. Bresenham.
Algorithm for computer control of a digital plotter.
IBM Systems Journal, 4(1) :25–30, 1965.
- [11] Jack E. Bresenham.
Incremental Line Compaction.
The Computer Journal, 25(1) :116–120, February 1982.
- [12] G. Bruhat.
Cours de physique générale - Optique.
Masson, 1992.

- [13] Stephen Cameron.
Efficient bounds in constructive solid geometry.
IEEE Computer Graphics and Applications, 11(3) :68–74, May 1991.
- [14] Thierry Carlier et Nicolas Szafran.
Structure de données pour la représentation et la manipulation des polyèdres.
Rapport de recherche RR 772-M, IMAG, Institut d’Informatique et de Mathématiques Appliquées de Grenoble, Mars 1989.
- [15] Frédéric Cazals, George Drettakis, et Claude Puech.
Filtering, Clustering and Hierarchy Construction : a New Solution for Ray-Tracing Complex Scenes.
Computer Graphics Forum, 14(3) :371–382, August 1995.
Proceedings of Eurographics ’95.
- [16] Mark J. Charney et Isaac D. Scherson.
Efficient Traversal of Well-Behaved Hierarchical Trees of Extents for Ray-Tracing Complex Scenes.
The Visual Computer, 6(3) :167–178, June 1990.
- [17] Jung-Hong Chuang et Weun-Jier Hwang.
A New Space Subdivision for Ray Tracing CSG Solids.
IEEE Computer Graphics and Applications, 15(6) :56–63, November 1995.
- [18] John G. Cleary et Geoff Wyvill.
Analysis of an algorithm for fast ray tracing using uniform space subdivision.
The Visual Computer, 4(2) :65–83, July 1988.
- [19] Daniel Cohen et Zvi Sheffer.
Proximity clouds - an acceleration technique for 3D grid traversal.
The Visual Computer, 11(1) :27–38, 1994.
- [20] Michael F. Cohen et John R. Wallace.
Radiosity and Realistic Image Synthesis.
Academic Press, Inc., London, 1993.
- [21] George E. Collins et Alkiviadis G. Akritas.
Polynomial real root isolation using descartes’rule of signs.
In *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation*, pages 272–275. ACM, 1976.
- [22] George E. Collins et Rudiger Loos.
Real zeros of polynomials.
Computing, Suppl. 4 :83–94, 1982.
- [23] Thomas Cormen, Charles Leiserson, et Ronald Rivest.
Introduction à l’algorithmique.
Dunod, Paris, 1994.
- [24] M. Daniel et J. C. Daubisse.
The numerical problem of using Bezier curves and surfaces in the power basis.
Computer Aided Geometric Design, 6(2) :121–128, 1989.
- [25] Robert Dautray, Jacques-Louis Lions, et Michel Artola.
Analyse mathématique et calcul numérique pour les sciences et les techniques : Évolution : numérique, transport, volume 9 of *CEA-INSTN, Collection enseignement*.
Masson, Paris, 1984.
- [26] Robert Dautray, Jacques-Louis Lions, et Michel Cessenat.
Analyse mathématique et calcul numérique pour les sciences et les techniques : Modèles physiques,

- volume 1 of *CEA-INSTN, Collection enseignement*.
Masson, Paris, 1984.
- [27] Jean-Pierre Dedieu et Jean-Claude Yakoubsohn.
Computing the real roots of a polynomial by the exclusion algorithm.
Numerical Algorithms, 4(1-2) :1-24, January 1993.
- [28] François-Régis Degott.
Partitionnement du plan par des courbes.
Thèse en mathématiques appliquées, Laboratoire LMC-IMAG, Université Joseph Fourier, Grenoble, Décembre 1991.
- [29] François-Régis Degott, Patrick Chenin, et Rémi Dessarce.
Algorithms and Data Structures for CAGD.
Congrès international Curves and Surfaces, Saint-Malo, Juillet 1999.
- [30] José Delfosse, William Terrence Hewitt, et Michel Mériaux.
An investigation of discrete ray tracing.
In *4th Discrete Geometry for Computer Imagery*, pages 65-76, Grenoble, Septembre 1994.
- [31] Ren DeLin.
Topics in Integral Geometry, volume 19 of *Series in Pure Mathematics*.
World Scientific, Singapore, New Jersey, London, Hong Kong, 1994.
- [32] Rémi Dessarce.
Calculs par lancer de rayons.
Thèse en mathématiques appliquées, Laboratoire LMC-IMAG, Université Joseph Fourier, Grenoble, Octobre 1996.
- [33] François Desvignes.
Rayonnements Optiques - Radiométrie - Photométrie.
Masson, Paris, 1991.
- [34] Olivier Devillers.
The Macro-Regions : an Efficient Space Subdivision Structure for Ray Tracing.
In W. Hansmann, F. R. A. Hopgood, et W. Strasser, editors, *Eurographics '89*, pages 27-38, Amsterdam, North-Holland, September 1989. Elsevier Science Publishers.
- [35] E. Durand.
Solutions numériques des équations algébriques - Équations du type $f(x)=0$, racines d'un polynôme, volume 1.
Masson, Paris, 1960.
- [36] Bruce E. Edwards.
Implementation of a Ray-Tracing Algorithm for Rendering Superquadrics Solids.
Master's thesis, Rensselaer Polytechnic Institute, Troy, December 1982.
- [37] Gerald Farin.
Curves and Surfaces for Computer Aided Geometric Design : a practical guide.
Academic Press, Inc., San Diego, London, 4 th edition, 1997.
- [38] R. T. Farouki et V. T. Rajan.
On the numerical condition of polynomials in Bernstein form.
Computer Aided Geometric Design, 4(3) :191-216, November 1987.
- [39] R. T. Farouki et V. T. Rajan.
Algorithms for polynomials in Bernstein form.
Computer Aided Geometric Design, 5(1) :1-26, June 1988.
- [40] Christine Froidevaux, Marie-Claude Gaudel, et Michèle Soria.
Types de données et algorithmes.
Collection Informatique. Ediscience international, Paris, 4ème edition, 1993.

- [41] Akira Fujimoto, Takayuki Tanaka, et Kansei Iwata.
ARTS : Accelerated Ray-Tracing System.
IEEE Computer Graphics and Applications, 6(4) :16–26, April 1986.
- [42] Michael Gervautz.
Three Improvements of the Ray Tracing Algorithm for CSG Trees.
Computer and Graphics, 10(4) :333–339, 1986.
- [43] Andrew S. Glassner.
Space Subdivision for Fast Ray Tracing.
IEEE Computer Graphics and Applications, 4(10) :15–22, October 1984.
- [44] Andrew S. Glassner.
Principles of Digital Image Synthesis, volume 2 of *The Morgan Kaufmann series in computer graphics and geometric modeling*.
Morgan Kaufmann Publishers, Inc., San Francisco, CA US, 1995.
- [45] Jeffrey Goldsmith et John Salmon.
Automatic Creation of Object Hierarchies for Ray Tracing.
IEEE Computer Graphics and Applications, 7(5) :14–20, May 1987.
- [46] Gérard Hacques.
Mathématiques pour l'informatique, volume 1.
Armand Colin, Paris, 1971.
- [47] Pat Hanrahan.
Ray Tracing Algebraic Surfaces.
Computer Graphics (SIGGRAPH '83 Proceedings), 17(3) :83–90, July 1983.
Held in Detroit, Michigan.
- [48] Ping-Kang Hsiung et Robert H. Thibadeau.
Accelerating ARTS.
The Visual Computer, 8(3) :181–190, March 1992.
- [49] Frederik W. Jansen.
Data structures for ray tracing.
In L. R. A. Kessener, F. J. Peters, et M. L. P. van Lierop, editors, *Data Structures for Raster Graphics*, pages 57–73. Springer-Verlag, New York, 1986.
Eurographic Seminar.
- [50] David Jevans et Brian Wyvill.
Adaptive Voxel Subdivision for Ray Tracing.
In *Proceedings of Graphics Interface '89*, pages 164–172, June 1989.
- [51] James T. Kajiya.
Ray Tracing Parametric Patches.
Computer Graphics (SIGGRAPH '82 Proceedings), 16(3) :245–254, July 1982.
Held in Boston, Massachusetts.
- [52] James T. Kajiya.
The Rendering Equation.
Computer Graphics (SIGGRAPH '86 Proceedings), 20(4) :143–150, August 1986.
Held in Dallas, Texas.
- [53] Devendra Kalra et Alan H. Barr.
Guaranteed Ray Intersections with Implicit Surfaces.
Computer Graphics (SIGGRAPH '89 Proceedings), 23(3) :297–306, July 1989.
Held in Boston, Massachusetts.

- [54] Michael R. Kaplan.
Space-Tracing : A Constant Time Ray-Tracer.
SIGGRAPH '85 State of the Art in Image Synthesis seminar notes, 18(3) :149–158, July 1985.
- [55] Timothy L. Kay et James T. Kajiya.
Ray Tracing Complex Scenes.
Computer Graphics (SIGGRAPH '86 Proceedings), 20(4) :269–278, August 1986.
Held in Dallas, Texas.
- [56] Hao-Ren Ke et Ruei-Chuan Chang.
An efficient hierarchical traversal algorithm for ray tracing.
The Visual Computer, 10(2) :79–87, 1993.
- [57] Mohammed Khachan.
Étude topologique pour la localisation et la reconstruction d'objets.
Thèse en mathématiques appliquées, Laboratoire LMC-IMAG, Université Joseph Fourier, Grenoble, 1998.
- [58] L. Christine Kinsey.
Topology of surfaces.
Undergraduate texts in mathematics. Springer-Verlag, New York, Berlin, Heidelberg, London, 1993.
- [59] Krzysztof S. Klimaszewski et Thomas W. Sederberg.
Faster Ray Tracing Using Adaptative Grids.
IEEE Computer Graphics and Applications, 17(1) :42–51, January-February 1997.
- [60] Pascal Lienhardt.
Topological Models for Boundary Representations : a Comparison with n -dimensional Generalized Maps.
Computer-Aided Design, 23(1) :59–82, 1991.
- [61] Pascal Lienhardt.
Aspects in Topology-Based Geometric Modeling.
Lecture Notes in Computer Science, 1347 :33–48, 1997.
- [62] Jean-Claude Léon.
Modélisation et construction de surfaces pour la CFAO.
Hermès, Paris, 1991.
- [63] J. David MacDonald et Kellogg S. Booth.
Heuristics for ray tracing using space subdivision.
The Visual Computer, 6 :153–166, 1990.
- [64] Denis Marchepoil.
Contribution à la modélisation géométrique : Équation algébriques, Modeleur, Lancer de rayons et Parallélisme.
Thèse en mathématiques appliquées, Laboratoire LMC-IMAG, Université Joseph Fourier, Grenoble, Février 1994.
- [65] Marie May.
Introduction à l'optique.
Dunod, Paris, 1993.
- [66] Donald Meagher.
Geometric Modeling Using Octree Encoding.
Computer Graphics and Image Processing, 19(2) :129–147, June 1982.
- [67] Aristides A. G. Requicha et Herbert B. Voelcker.
Constructive Solid Geometry.

- Technical Report Technical Memo 25, Production Automation Project, University of Rochester, Rochester, NY, 1977.
- [68] Marc Roelens.
Un environnement pour le tracé de rayons utilisant une modélisation par arbre de construction.
Thèse en informatique, École Nationale Supérieure des Mines, Saint-Etienne, Avril 1993.
- [69] S. D. Roth.
Ray Casting for Modelling Solids.
Computer Graphics and Image Processing, 18 :109–144, February 1982.
- [70] Steven M. Rubin et Turner Whitted.
A 3-Dimensional Representation for Fast Rendering of Complex Scenes.
Computer Graphics (SIGGRAPH '80 Proceedings), 14(3) :110–116, July 1980.
Held in Seattle, Washington.
- [71] Walter Rubin.
Analyse réelle et complexe.
Masson, Paris, Milan, Barcelone, 1975.
Traduit de l'américain par N. Dhombres et F. Hoffman.
- [72] Luis A. Santoló.
Introduction to Integral Geometry.
Actualités Scientifiques et Industrielles. Hermann & Cie, Paris, 1953.
Publications de l'Institut de Mathématique de l'université de Nancago.
- [73] Luis A. Santoló.
Integral Geometry and Geometric Probability.
Addison-Wesley, Reading, London, Amsterdam, 1976.
- [74] Isaac D. Scherson et Elisha Caspary.
Data structures and the time complexity of ray tracing.
The Visual Computer, 3(4) :201–213, December 1987.
- [75] T. W. Sederberg et D. C. Anderson.
Ray Tracing of Steiner Patches.
Computer Graphics (SIGGRAPH '84 Proceedings), 18(3) :159–164, July 1984.
Held in Minneapolis, Minnesota.
- [76] Thomas Warren Sederberg.
Implicit and Parametric Curves and Surfaces for Computer Aided Geometric Design.
Phd thesis, Purdue University, West Lafayette, August 1983.
- [77] Robert Siegel et John R. Howell.
Thermal Radiation Heat Transfert.
Hemisphere Publishing Corporation, Washington, London, third edition, 1992.
- [78] François Xavier Sillion et Claude Puech.
Radiosity and Global Illumination.
Morgan Kaufmann Publishers, Inc., San Francisco, CA US, 1994.
- [79] Mel Slater.
Tracing a ray through uniformly subdivided n-dimensional space.
The Visual Computer, 9(1) :39–46, October 1992.
- [80] John M. Snyder et Alan H. Barr.
Ray Tracing Complex Models Containing Surface Tessellations.
Computer Graphics (SIGGRAPH '87 Proceedings), 21(4) :119–128, July 1987.
Held in Anaheim, California ; 27-31 July 1987.

-
- [81] H. A. Steinberg.
A Smooth Surface Based on Biquadratic Patches.
IEEE Computer Graphics and Applications, 4(9) :20–23, September 1984.
- [82] Michael A. J. Sweeney et Richard H. Bartels.
Ray tracing free-form B-spline surfaces.
IEEE Computer Graphics and Applications, 6(2) :41–49, February 1986.
- [83] Daniel L. Toth.
On Ray Tracing Parametric Surfaces.
Computer Graphics (SIGGRAPH '85 Proceedings), 19(3) :171–179, July 1985.
Held in San Francisco, California.
- [84] Hank Weghorst, Gary Hooper, et Donald P. Greenberg.
Improved Computational Methods for Ray Tracing.
ACM Transactions on Graphics, 3(1) :52–69, January 1984.
- [85] Kyu-Young Whang, Ju-Won Song, Ji-Woong Chang, Ji-Yun Kim, Wan-Sup Cho, Chong-Mok Park, et Il-Yeol Song.
Octree-R : An Adaptive Octree for Efficient Ray Tracing.
IEEE Transactions on Visualization and Computer Graphics, 1(4) :343–349, December 1995.
- [86] Turner Whitted.
An Improved Illumination Model for Shaded Display.
Communications of the ACM, 23(6) :343–349, June 1980.
- [87] Geoff Wyvill et Toshiyasu L. Kunii.
A Functional Model for Constructive Solid Geometry.
The Visual Computer, 1(1) :3–14, July 1985.
- [88] Geoff Wyvill, Toshiyasu L. Kunii, et Y. Shirai.
Space Subdivision for Ray Tracing in CSG.
IEEE Computer Graphics and Applications, 6(4) :28–34, April 1986.
- [89] Roni Yagel, Daniel Cohen, et Arie Kaufman.
Discrete Ray Tracing.
IEEE Computer Graphics and Applications, 12(5) :19–28, September 1992.