

UNIVERSITÉ TOULOUSE III - PAUL SABATIER
MASTER 2 - INFORMATIQUE GRAPHIQUE ET ANALYSE D'IMAGES

NATIONAL INSTITUTE OF INFORMATICS
TOKYO, JAPON

MÉMOIRE - STAGE DE FIN D'ÉTUDES

**Rendering of fluorescent materials using
spectral path tracing : Niixtracer, a custom
rendering engine**

Nolan MESTRES
16 Mars - 29 Août 2018

Encadrant universitaire
Pr. Mathias PAULIN



Directrice de stage
Pr. Imari SATO



Remerciements

Pour l'accueil au sein de la division de recherche Digital Content and Media Sciences du National Institute of Informatics, son expérience, les pistes et conseils précieux prodigués, ma directrice de stage, le Pr. Imari Sato.

L'ensemble de l'équipe de recherche du Pr. Imari Sato et plus particulièrement Ilyes Kacher, assistant de recherche au sein de l'équipe, pour son aide et sa bonne humeur, mais également pour son sens de l'orientation et ses bonnes adresses à Tokyo.

Agata Soccini, la meilleure élève de Japonais à qui il m'a été donné d'enseigner, pour son amitié et sa sagesse.

Pour son amitié également, sa maîtrise de la langue japonaise et sa patience, Célia Boffo, sans qui les démarches administratives et autres problèmes logistiques auraient été bien plus complexes.

Pour sa bienveillance, son énergie et son investissement personnel pour ses élèves actuels comme anciens, Laurence Bergamini.

Enfin, pour leurs oreilles attentives et leur soutien sans faille dans cette expérience, les jours de soleil comme de pluie, Joëlle Mestres et Manon Boschard.

Table des matières

1 Contexte du stage	5
1.1 Tokyo, Japon	5
1.2 National Institute of Informatics	6
1.3 Une approche différente de la recherche	7
2 Problématique et contributions	8
2.1 Synthèse d'images et rendu physiquement réaliste	8
2.2 Définition du sujet et objectifs	9
2.2.1 Évolutions du sujet	9
2.3 Contributions	10
2.3.1 Moteur de rendu	10
2.3.2 Rendu de liquides fluorescents	10
3 État de l'art	11
3.1 Équation du rendu	11
3.1.1 Forme générale	12
3.1.2 Bidirectional Reflectance Distribution Function	12
3.1.3 Limites	13
3.2 Tracé de rayons	14
3.2.1 Concept	14
Définition mathématique des objets	14
Extension de l'algorithme	14
3.2.2 Path tracing	15
Méthode de Monte Carlo	15
Méthodes d'échantillonnage	16
Roulette russe	17
Modèles de BRDF	18
3.3 Colorimétrie	18
3.3.1 Distribution Spectrale de Puissance	18
3.3.2 Perception humaine de la couleur	19
3.3.3 Du spectre à la couleur du pixel	19
Mise en correspondance tonale	20
Correction gamma	20
3.4 Fluorescence	20
3.4.1 Déplacement de Stokes	20
3.4.2 Modèle de Glassner	21
4 Niixtracer, un moteur de rendu physiquement réaliste	23
4.1 Fonctionnalités	23
4.1.1 Fichiers de scène	23
Qt XML	23

	Formes géométriques, plans et sphères	24
4.1.2	Ray tracing	24
	Réflexions et réfractions	24
	Ombres	24
	Nombre de rebonds d'un rayon	25
	Antialiasing	25
4.1.3	Path tracing	26
	Bidirectional Reflectance Distribution Functions	26
	Échantillonnage explicite des sources	26
4.1.4	Spectral path tracing	28
	Distributions Spectrales de Puissance	29
	Illuminants	29
4.1.5	Fluorescence	31
	Distribution de Cauchy asymétrique	31
4.2	Résultats	32
4.2.1	Nombre d'échantillons	32
4.2.2	Performances	33
5	Travaux futurs	35
5.1	Rendu de liquides fluorescents	35
5.2	Iridescence	35
6	Bilans technique et personnel	37
6.1	Bilan technique	37
6.1.1	Pour le laboratoire	37
6.1.2	Compétences acquises	37
6.2	Bilan personnel	38
6.3	Perspectives d'avenir	38
7	Conclusion	40
	Annexes	41
A	Fichiers de scène	42
B	Ray tracing	46
C	Path tracing	48
D	Spectral path tracing	50
E	Fluorescence	52

Partie 1

Contexte du stage

Ce mémoire s'inscrit dans le cursus du Master Informatique Graphique et Analyse d'Images de l'université Toulouse III Paul Sabatier. Il s'appuie sur le travail que j'ai effectué lors du stage de fin d'année de Master 2, dans un laboratoire de recherche japonais, le National Institute of Informatics (NII) à Tokyo.

Cette première partie s'attache à présenter le contexte du stage et donc pour commencer l'environnement dans lequel j'ai évolué pendant ces six mois. Il n'est pas ici directement question du travail effectué au cours du stage, néanmoins je trouve ces informations pertinentes et nécessaires à la bonne compréhension de celui-ci et du monde de la recherche au Japon, et plus particulièrement à Tokyo.

1.1 Tokyo, Japon

Avant d'étudier l'informatique, j'ai poursuivi à l'université des études de Langue, Littérature et Civilisation Étrangère (LLCE), mention Japonais. C'est donc très tôt après ma réorientation que je me suis renseigné sur les possibilités de stage à l'étranger et notamment au Japon. Il se trouve que l'Institut de Recherche en Informatique de Toulouse (IRIT) possède un partenariat avec le NII à Tokyo. C'est ainsi qu'après avoir passé les différentes étapes de sélection j'ai eu la chance d'effectuer mes six mois de stage dans ce cadre si particulier.

Tokyo en quelques chiffres, c'est plus de 39 millions d'habitants pour une superficie vingt fois supérieure à celle de Paris. La densité de population y est donc étonnamment bien plus faible. Cependant, il est facile de s'y sentir écrasé par les gratte-ciels et certains quartiers restent bondés à toute heure du jour comme de la nuit.

Heureusement, proche du site sur lequel j'ai travaillé, dans l'antenne principale du laboratoire, se trouvent quelques espaces verts et notamment le palais impérial et son jardin.

Les transports et loyers étant les deux budgets les plus élevés de la vie tokyoïte, il a fallu choisir entre un long temps de trajet avec appartement d'une dizaine de mètres carrés ou bien une chambre minuscule proche du lieu de travail. J'ai opté pour la seconde option mais je ne saurais toujours pas dire si j'ai fait le bon choix.



FIGURE 1.1 – Vues de Tokyo. À gauche, vue lointaine du quartier du luxe, Ginza, à droite le quartier populaire Coréen Shin-Ōkubo

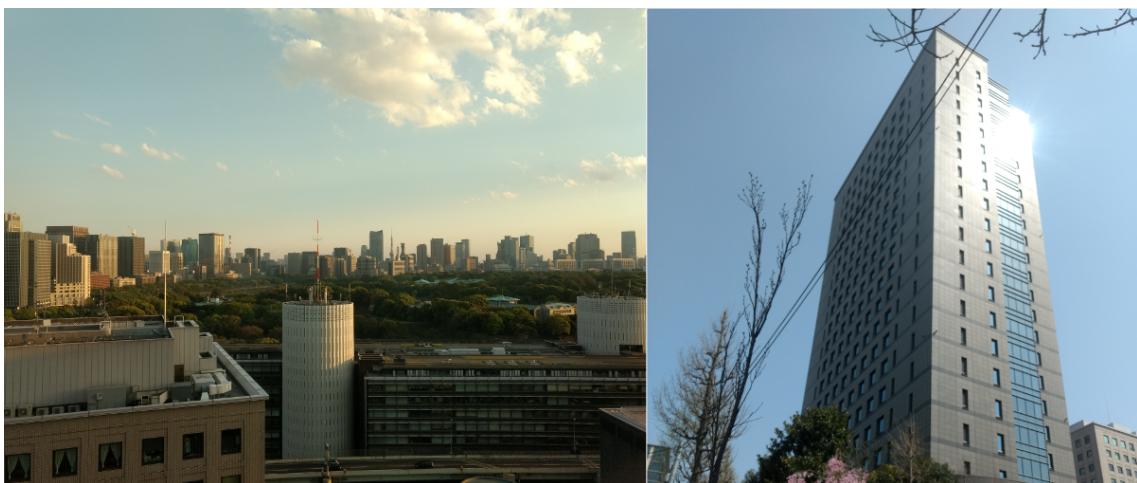


FIGURE 1.2 – À gauche, vue depuis le 18ème étage du laboratoire, à droite le bâtiment en lui-même.

1.2 National Institute of Informatics

Une fois le dilemme logement-transport résolu, j'ai pu commencer mon travail au sein de l'équipe dirigée par le professeur Imari Sato dans la division Digital Content and Media Sciences du laboratoire. L'équipe comporte plusieurs doctorants, assistants de recherche et post-doctorants, pour un total de 10 personnes. Bien que composée de Japonais et d'étrangers, je n'ai eu à échanger qu'avec des non-Japonais. En effet, nous ne travaillions pas dans les mêmes bureaux. Ce que je pensais être une bonne occasion de pratiquer mon Japonais s'est avéré infondé, et bien qu'un peu décevant, il faut avouer que la communication en Anglais était malgré tout bien plus aisée.

Le principal domaine de recherche de l'équipe est la vision par ordinateur (notamment les problèmes inverses de reconstruction de formes) dont l'état de l'art est de nos jours fortement teinté de Machine Learning. C'est cette direction que prend bon nombre de membres de l'équipe. De mon côté, après m'être renseigné avant mon arrivée sur les travaux de ma superviseure, j'ai voulu orienter mon sujet vers une autre discipline, l'informatique graphique, et plus spécifiquement le rendu. En effet, le professeur Imari Sato

ayant beaucoup travaillé par le passé sur la fluorescence et étant moi-même très intéressé par cet effet visuel, nous avons convenu qu'il serait intéressant d'élargir le domaine de recherche de l'équipe.

1.3 Une approche différente de la recherche

Lors des différents aperçus que j'ai pu avoir du monde de la recherche en Europe, que ce soit à Toulouse à l'université Paul Sabatier lors du Travail Encadré de Recherche en Licence 3, en Master lors des différents projets, ou encore en Pologne à l'université de Sciences et Technologie de Cracovie, l'approche était similaire. Ici, j'ai été assez surpris de constater que les travaux sont rarement individuels (ce en quoi mon stage fait un peu exception, travaillant seul), et c'est le groupe qui va primer sur l'individu (quitte à ajouter des co-auteurs aux publications sans raison apparente).

Néanmoins, même si les retombées d'un travail doivent rejoindre sur l'ensemble du groupe, le cadre reste très libre et l'autonomie est une qualité *sine qua non* pour réussir. Dans mon cas, j'étais totalement libre de l'orientation de mon travail, jusqu'au sujet lui-même. Bien que très appréciable, cette liberté dans le travail avait aussi de quoi effrayer compte tenu de mon manque d'expérience à mener des recherches seul et des enjeux et attentes qui pesaient sur mes épaules. Je ne voudrais pas véhiculer de fausses généralités mais d'après mes collègues (principalement de nationalité Chinoise et Thaïlandaise), ce que j'ai pu constater au Japon sur l'importance du groupe peut se généraliser assez systématiquement à toute l'Asie.

Partie 2

Problématique et contributions

Dans cette partie nous allons dans un premier temps présenter le domaine de la synthèse d'images et plus particulièrement celui du rendu physiquement réaliste, puis le sujet et objectifs du stage ainsi que leurs évolutions. Enfin les contributions apportées à l'issue de ce travail seront passées en revue.

2.1 Synthèse d'images et rendu physiquement réaliste

La synthèse d'images, ou encore le rendu (du terme anglais *rendering*), est une discipline de l'informatique graphique dont le but est de générer une image en deux dimensions à l'aide d'un programme informatique, appelé moteur de rendu dans le cas qui nous intéresse. Ce moteur de rendu prendra en entrée la description d'une scène 3D et produira en sortie l'image de synthèse correspondante. Le rendu ainsi obtenu peut être photoréaliste ou non-photoréaliste (Figure 2.1), selon la technique utilisée. Ces deux types de rendu, le rendu physiquement réaliste (*physically based rendering* ou PBR en anglais) d'un côté et le rendu expressif de l'autre, répondent à des problématiques très différentes.

En ce qui nous concerne, ce mémoire s'inscrit dans le cadre du rendu physiquement réaliste. Le but ultime d'une image de synthèse issue d'un processus de rendu photoréalistes.



(a) Rendu de diamants avec réflexions et réfractions par Josh Wiseman.



(b) Rendu pictural pour de l'animation, projet de Barbara Meier pour Disney.

FIGURE 2.1 – Exemples de rendu physiquement réaliste (2.1a) et de rendu expressif (2.1b).

liste est d'être impossible à discerner, par un observateur humain, d'une photographie du monde réel. Pour cela, le rendu physiquement réaliste se base sur différentes disciplines, notamment l'optique (branche de la physique ayant trait à la lumière), mais aussi les mathématiques (algèbre linéaire, probabilités, intégration numérique, trigonométrie, etc.), la colorimétrie, et bien sûr l'informatique. Un des ouvrages de référence au sujet du rendu physiquement réaliste dans son ensemble, sur lequel je me suis beaucoup appuyé au cours de ce stage, est le livre *Physically Based Rendering : From Theory to Implementation* de Matt Pharr et Greg Humphreys [PH10].

Plus précisément, les images de synthèse que nous avons générées au cours de ce travail sont le fruit d'un processus de rendu hors-ligne (en opposition au rendu temps-réel). Les domaines d'application du rendu hors-ligne se trouvent principalement dans l'industrie cinématographique, où l'usage d'effets spéciaux numériques (*Computer-Generated Imagery, CGI*) est désormais omniprésent. De même, les images des films d'animation sont générées dans d'immenses fermes de calcul pendant plusieurs mois. De nombreuses cinématiques de jeux-vidéo sont également ainsi précalculées. Mais le domaine du divertissement n'est pas le seul à recourir au rendu photoréaliste. Il est également utilisé en industrie, pour de la simulation précise, en architecture pour certaines étapes de projets où les illustrations et dessins au trait ne suffisent pas, mais aussi en recherche où des rendus de référence sont parfois nécessaires pour vérifier des résultats, effectuer des calculs d'erreur, faire de la reconstruction 3D, etc.

2.2 Définition du sujet et objectifs

À mon arrivée au Japon, le sujet n'était pas clairement défini car tout était resté très ouvert lors des échanges préalables que j'ai pu entretenir avec ma superviseure. Contrairement aux sujets émis par des entreprises en France que j'ai pu consulter ou même ceux de l'Institut de Recherche en Informatique de Toulouse rattaché à mon université, il ne m'a jamais été formulé un sujet avec contexte et problématique. J'étais donc assez décontenancé lors de cette première semaine au laboratoire, car personne ne me donnait de directives ou de pistes.

J'ai donc compris que l'initiative devait venir de mon côté et que si j'attendais qu'un sujet net et précis me soit préparé, rien ne bougerait. Ce dont j'étais certain était que je voulais travailler sur du rendu et la fluorescence. J'ai donc préparé une présentation sur le sujet pour l'équipe, à l'issue de laquelle ma superviseure, le Pr. Imari Sato, et moi-même avons convenu qu'il serait intéressant pour l'équipe de développer un moteur de rendu en interne. Ce moteur devrait reprendre des caractéristiques des systèmes physiquement réalisistes et permettre au laboratoire d'effectuer des rendus hors-ligne de scènes 3D. Jusqu'alors le laboratoire utilisait des moteurs de rendu open-sources tels que Mitsuba ou PBRT. Ces systèmes, bien que documentés, ne sont néanmoins pas simples à prendre en main et il est encore moins évident de les modifier pour quelqu'un qui ne viendrait pas de la communauté de l'informatique graphique (ce qui n'est pas le cas de la majorité des membres de l'équipe).

2.2.1 Évolutions du sujet

Ne sachant pas vraiment à quoi m'attendre en entreprenant seul le développement d'un moteur de rendu physiquement réaliste, j'ai procédé par étapes successives. Les

objectifs ont donc évolué tout au long du stage en se raffinant et en se complexifiant. Je suis parti d'un fichier texte vide pour arriver à un simple système de tracé de rayons faisant beaucoup d'approximations et de simplifications, puis à un système plus complet, prenant en compte les effets optiques classiques (réflexions, réfractions), le *color bleeding*¹, les caustiques, le rendu spectral, la fluorescence, etc. Nous reviendrons plus tard à ces différentes fonctionnalités, à leurs fonctionnements et significations. Au final, j'ai donc bel et bien développé un moteur de rendu physiquement réaliste, dans son intégralité.

C'est par ces évolutions successives que l'intitulé de mon stage est devenu : « **Rendu de matériaux fluorescents au moyen d'un algorithme de path tracing spectral : Niixtracer, un moteur de rendu personnalisé.** »

2.3 Contributions

2.3.1 Moteur de rendu

Vous l'aurez compris, la contribution principale qui a été mise en place au cours de ces six mois est un moteur de rendu physiquement réaliste permettant le rendu de scènes 3D. Ces scènes 3D sont décrites finement par l'utilisateur dans des fichiers de scènes séparés, permettant une grande liberté dans la création de celles-ci. Les objets qu'elles contiennent peuvent être composés de différents matériaux, fluorescents ou non. Les scènes décrites peuvent avoir différents illuminants, comme nous le verrons par la suite.

Le moteur en lui-même, développé en C++ en partant de zéro, est un vrai plus pour l'équipe car il permettra dans le futur de fournir de nombreuses images de référence pour résoudre des problèmes inverses en vision par ordinateur. Il permettra aussi aux autres membres de l'équipe, actuels et futurs, de reprendre ces travaux et de les améliorer, et pourquoi pas d'en retirer des publications.

C'est donc aussi une ouverture pour l'équipe sur l'informatique graphique et le rendu, avec peut-être à la clef des projets et collaborations interdisciplinaires, le NII ayant un partenariat direct avec l'IRIT.

2.3.2 Rendu de liquides fluorescents

Enfin, une autre contribution que nous avons évoquée et sur laquelle nous avons commencé à travailler est celle du rendu de liquides fluorescents. Pour cela il faut enrichir le moteur déjà existant pour lui permettre de rendre des milieux participatifs² [Rus88]. Ce travail en cours est basé, entre autres, sur l'article de Marwann Abdellah et al. paru en Juin 2017 [Abd+17]. Pour le moment cette fonctionnalité n'est pas opérationnelle, mais une base est présente.

1. La couleur d'un objet affecte celles des autres objets proches, il s'agit d'inter-réflexions diffuses dues à l'illumination indirecte.

2. On parle de milieu participatif quand le milieu dans lequel évolue la lumière a un effet sur elle : par exemple des interactions entre les photons et des particules en suspension dans le milieu comme cela peut être le cas pour la fumée, des liquides comportant des impuretés, etc.

Partie 3

État de l'art

Dans cette troisième partie et les sections qui la composent, nous présentons et détaillons les travaux majeurs du domaine qui ont rendu possible le travail décrit dans ce mémoire. Le domaine du rendu hors-ligne étant vaste et relativement ancien à l'échelle de l'informatique nous ne décrirons pas ici l'intégralité des articles liés à nos travaux mais seulement ceux sur lesquels nous nous sommes directement appuyés ou qui ont eu un impact crucial sur la synthèse d'images photoréalistes.

3.1 Équation du rendu

C'est l'équation intégrale qui décrit les transferts radiatifs dans une scène (équation 3.1). Dans notre cas, elle décrit l'interaction des photons avec la matière (absorption, émission, diffusion), c'est-à-dire la luminance énergétique (ou radiance, terme que nous allons préférer tout au long du mémoire) quittant un point d'une surface.

C'est cette équation que tentent de résoudre les différents algorithmes de rendu. Elle a été simultanément introduite pour la première fois en 1986 lors de la conférence SIGGRAPH de Dallas aux États-Unis, par David Immel et al. [ICG86], ainsi que par James T. Kajiya [Kaj86].

Elle s'appuie sur le principe physique de la conservation de l'énergie : autrement dit que la distribution de la radiance dans la scène est à l'état d'équilibre. La radiance est la puissance du rayonnement émis ou propagé en un point d'une surface, par unité de surface, par unité d'angle solide (stéradian). L'unité du système international d'unités pour la radiance est donc le $W.m^{-2}.sr^{-1}$.

Ainsi, ce principe de conservation de l'énergie établit que l'énergie qui quitte un point d'une surface est égale à la somme de l'énergie émise et de l'énergie réfléchie depuis ce point.

3.1.1 Forme générale

Dans sa forme la plus courante, l'équation du rendu peut s'écrire comme suit :

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{\Omega} f_r(p, \omega_o, \omega_i) L_i(p, \omega_i) (\omega_i \cdot \vec{n}) d\omega_i \quad (3.1)$$

Avec :

- $L_o(p, \omega_o)$ la radiance quittant le point p , dans la direction ω_o ;
- $L_e(p, \omega_o)$ la radiance émise en p , dans la direction ω_o ;
- $L_i(p, \omega_i)$ la radiance arrivant en p , depuis la direction ω_i ;
- Ω le domaine d'intégration des directions ω_i , correspondant à l'hémisphère unitaire centrée en \vec{n} ;
- $f_r(p, \omega_o, \omega_i)$ la fonction de distribution probabiliste de réflectivité bidirectionnelle (*Bidirectional Reflectance Distribution Function*, BRDF, en anglais.) que nous allons expliquer dans la section suivante ;
- $(\omega_i \cdot \vec{n})$ le produit scalaire entre la normale \vec{n} en p et la direction incidente de la lumière ω_i . Ce terme vient de la loi en cosinus de Lambert qui dit que la radiance observée en un point d'une surface est directement proportionnelle au cosinus de l'angle entre la direction de la lumière incidente et la normale à la surface en ce point.

3.1.2 Bidirectional Reflectance Distribution Function

Les *Bidirectional Reflectance Distribution Functions* (BRDF) sont des fonctions qui permettent de spécifier la réflectivité d'une surface opaque de manière non-ambiguë, selon la géométrie des rayons lumineux qui la frappent. C'est à dire qu'une BRDF est fonction de la direction de la lumière incidente, et de celle de la lumière réfléchie (voir Figure 3.1). Elles permettent de définir un ratio entre la radiance réfléchie dans une direction ω_o et l'irradiance incidente depuis une direction ω_i (Équation 3.2) [Nic65].

$$f_r(p, \omega_r, \omega_i) = \frac{dL_r(p, \omega_r)}{dE_i(p, \omega_i)} = \frac{dL_r(p, \omega_r)}{L_i(p, \omega_i) \cos \theta_i d\omega_i} \quad (3.2)$$

Elles sont régies par des principes physiques :

- la réciprocité de Helmholtz, qui dit que pour toutes paires de directions ω_r et ω_i on a :

$$f_r(p, \omega_r, \omega_i) = f_r(p, \omega_i, \omega_r)$$

Cela implique qu'il serait possible d'inverser source et capteur dans une scène. Le résultat de la mesure du flux énergétique (grandeur qui décrit la puissance du rayonnement électromagnétique, en watt) resterait inchangé.

- la conservation de l'énergie, qui dit que la radiance réfléchie est inférieure ou égale à la radiance incidente, quelque soit la direction de la lumière incidente considérée. D'où :

$$\int_{\Omega} f_r(p, \omega_r, \omega_i) \cos \theta_r d\omega_r \leq 1$$

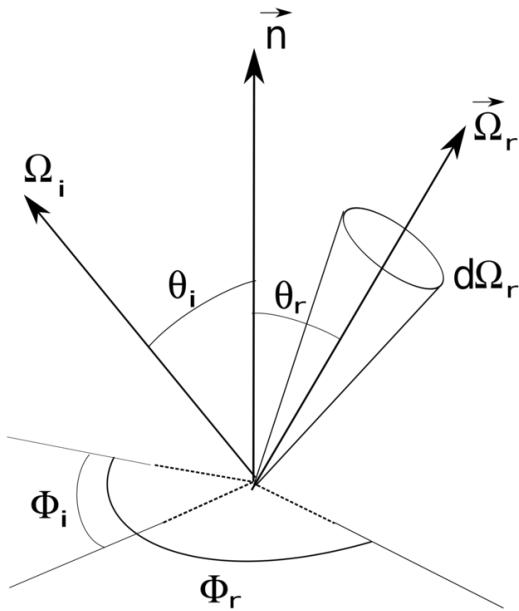


FIGURE 3.1 – Définition et représentation des termes qui composent la BRDF

Inférieur ou égal et non pas égal. Cela semble contre-intuitif compte-tenu du fait qu'il est question de conservation de l'énergie, mais il faut ne pas perdre de vue qu'une des interactions possibles des photons avec la matière reste l'absorption d'une partie de l'énergie par celle-ci, et donc la possibilité que toute la radiance incidente ne soit pas réfléchie.

Nous avons maintenant en notre possession toutes les clefs pour comprendre l'équation du rendu (Équation 3.1) et ses limites.

3.1.3 Limites

Bien que modélisant bon nombre de phénomènes visuels du monde réel, tels que le *color bleeding*, les caustiques, les effets de pénombre, etc., l'équation du rendu n'est néanmoins pas dépourvue de limites. James T. Kajiya en était d'ailleurs bien conscient car il en cite certaines dans son article. Dans sa forme générale telle que donnée précédemment, elle est, je cite, « une approximation aux équations de Maxwell [...] basée sur l'optique géométrique. ». En effet, aucune mention n'est faite du temps, de la longueur d'onde des photons, la radiance est considérée invariante le long d'un rayon lumineux, etc.

Ainsi, cette équation ne permet pas de rendre des phénomènes pourtant courants et déterminants pour atteindre l'objectif fixé du photoréalisme. Par exemple :

- les milieux participatifs et donc l'impact des milieux sur les changements de radiance le long des rayons lumineux;
- les différences de reflectivité d'une surface en fonction du spectre de la lumière qui la frappe;
- la fluorescence, où la lumière absorbée à une certaine longueur d'onde est réémise à une longueur d'onde plus élevée;
- la phosphorescence, où la lumière est absorbée à un moment t , avant d'être ré-

émise après un laps de temps de l'ordre de 10 nanosecondes, parfois à une longueur d'onde différente ;
— etc.

C'est pourquoi les algorithmes qui tentent de résoudre l'équation du rendu ont fait l'objet de nombreux travaux au cours des trente dernières années. L'équation en elle-même a été étendue à plusieurs reprises et améliorée pour combler ces lacunes, avec succès au vu des rendus qu'il est maintenant possible d'effectuer !

3.2 Tracé de rayons

Maintenant que nous avons présenté l'équation du rendu (Équation 3.1) nous allons aborder une classe d'algorithmes utilisée pour la résoudre (ou tout du moins en approximer la solution) : les algorithmes de tracé de rayons.

3.2.1 Concept

Les algorithmes de tracé de rayons (*ray tracing*) sont basés sur l'optique géométrique, qui permet de modéliser le comportement de la lumière. C'est une approximation de la nature réelle de la lumière¹ mais qui permet néanmoins de rendre compte d'une importante partie des phénomènes optiques (réflexion, réfraction,...) observables au quotidien.

Le principe fondamental (initialement introduit dans [App68]) consiste à lancer des rayons (*ray casting*) dans une scène 3D. Le lancer d'un rayon permet de récupérer des informations sur le premier objet intersecté par celui-ci, informations qui vont nous servir pour déterminer la couleur des pixels de l'image à rendre. L'origine des rayons est l'œil de l'observateur, ou encore une caméra placée dans la scène. Un plan image est défini entre l'observateur et la scène, et des rayons lancés à travers les pixels de celle-ci. Chaque pixel prendra pour couleur celle de l'objet intersecté par le rayon qui le traverse. Avec cette méthode, seule les surfaces visibles sont prises en compte dans le calcul de la couleur (on parle d'éclairage direct).

Définition mathématique des objets

Les surfaces des objets qui composent la scène sont définies mathématiquement. Pour vérifier si le rayon intersecte un objet dans la scène il faut parcourir l'ensemble des objets et résoudre les équations de leurs surfaces pour ce rayon. C'est un travail mathématique complexe et c'est cette étape qui va monopoliser notre temps de calcul pendant le rendu d'une image, c'est pourquoi l'optimiser est très important.

Extension de l'algorithme

Cet algorithme simple va ensuite être étendu par Turner Whitted pour prendre en compte les réflexions, réfractions et les ombres (Figure 3.2). Les calculs ne s'arrêtent donc

1. Nature duale, ondulatoire et corpusculaire.

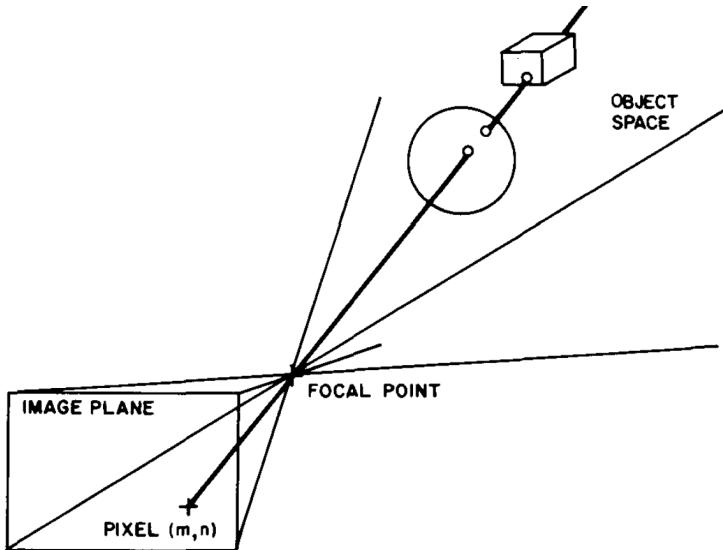


FIGURE 3.2 – Figure issue de [Whi80] montrant le principe du tracé de rayons

plus à la première surface rencontrée par le rayon mais sont réfléchis ou transmis (appel récursif du lancer de rayon) selon les lois de l'optique géométrique (Snell-Descartes).

Ici, le choix est fait de ne pas tracer des rayons depuis les sources lumineuses, comme le ferait la lumière en réalité. En effet, espérer qu'ils rebondissent jusqu'à atteindre le capteur du dispositif capturant l'image est très peu probable et pourrait nécessiter une infinité de rebonds.

Le modèle d'illumination utilisé (*shading model*, en anglais) est fortement inspiré du modèle Blinn-Phong [Bli77]. Bien qu'à l'origine des images de synthèse les plus réalistes de l'époque (fin des années 70, début des années 80) ce modèle reste néanmoins très limité.

C'est en 1986, en même temps qu'il présente l'équation du rendu, que Kajiya va proposer un algorithme qui calcule une solution numérique non biaisée de celle-ci. C'est le *path tracing* [Kaj86].

3.2.2 Path tracing

Méthode de Monte Carlo

Le path tracing fournit une solution approximée à l'équation du rendu en s'appuyant sur la méthode de Monte Carlo, une méthode stochastique pour résoudre des problèmes d'intégration numérique. Elle tend vers la solution exacte du problème.

Le principe est de prendre un grand nombre d'échantillons de notre variable aléatoire, d'évaluer l'intégrale pour ces valeurs et de faire la moyenne des résultats pour obtenir une approximation de l'intégrale. L'erreur commise est fonction du nombre d'échantillons tirés, mais la convergence est néanmoins lente : soit N le nombre d'échantillons, augmenter N va réduire l'erreur de \sqrt{N} .

Soit I l'intégrale de $f(x)$ sur Ω ,

$$I = \int_{\Omega} f(x) dx$$

Alors, pour $pdf(x)$ une densité de probabilité définie sur Ω ,

$$I = \int_{\Omega} f(x) dx = \int_{\Omega} \frac{f(x)}{pdf(x)} pdf(x) dx$$

On a donc,

$$I = E\left[\frac{f(X)}{pdf(X)}\right] = \lim_{N \rightarrow +\infty} \frac{1}{N} \sum_{n=1}^N \frac{f(x_i)}{pdf(x_i)}$$

Avec $E[Y]$ l'espérance de Y , et X une variable aléatoire de densité de probabilité pdf . Calculer la valeur exacte de I n'est donc pas possible, seulement une estimation telle que :

$$I \approx \frac{1}{N} \sum_{n=1}^N \frac{f(x_i)}{pdf(x_i)}$$

C'est cette méthode qui va nous permettre de calculer une estimation de la solution de l'équation du rendu. C'est un outil très puissant car il permet en l'occurrence de résoudre numériquement des problèmes intégraux de dimension infinie dont il est inconcevable de trouver une solution en forme close.

Pour ce qui est de la littérature sur le sujet, je me suis principalement basé sur les supports de cours du professeur Mathias Paulin [Pau17], mon encadrant universitaire pour ce mémoire, et leurs références, avec entre autres, l'ouvrage de Hammersley et Handscomb [HH65].

Méthodes d'échantillonnage

Le path tracing étant un algorithme basé sur les méthodes de Monte Carlo, nous sommes amenés à effectuer de nombreux tirages probabilistes. Plutôt que d'utiliser une distribution uniforme, par exemple pour déterminer dans quelle direction de l'hémisphère centrée sur la normale \vec{N} en p (Figure 3.3) un rayon rebondi, il est possible d'avoir recours à des méthodes d'échantillonnage plus efficaces. Parmi ces méthodes se trouve notamment l'échantillonnage stratifié ou bien les suites à discrépance² faible (*low-discrepancy sequences*). De telles méthodes d'échantillonnage sont préférées à une distribution uniforme classique car elles assurent la bonne répartition des échantillons sur le domaine considéré. La variance s'en trouve réduite, ce qui nous permet donc, à nombre d'échantillons identique, d'augmenter la précision des rendus.

Dans le cas plus particulier des suites à discrépance faible (séquence de Sobol, séquence de Halton, etc.), le terme de suites quasi-aléatoires est avancé. Elles sont en réalité déterministes ce qui assurent une cohérence du bruit entre les différents rendus générés. Cela peut s'avérer utile pour créer une animation à partir de plusieurs images de

2. La discrépance est l'écart maximal entre les échantillons tirés dans un domaine par rapport à des échantillons qui seraient équirépartis sur ce même domaine.

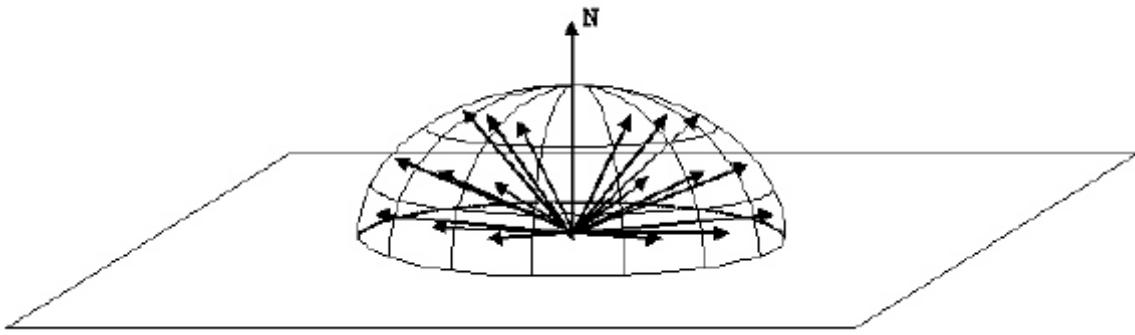


FIGURE 3.3 – Échantillonnage de l'hémisphère en un point p de la surface, de normale \vec{N}

synthèse, où la cohérence du bruit haute fréquence entre les différentes images de l'animation a son importance. De plus, la convergence d'une méthode dite de quasi-Monte Carlo est supérieure ($O(\frac{1}{N})$ contre $O(N^{-0.5})$).

Roulette russe

L'équation du rendu étant récursive, il faut trouver un critère d'arrêt satisfaisant. Comme nous venons de le voir, la méthode de Monte Carlo est non-biaisée, mais arrêter les réflexions des rayons à une profondeur donnée introduirait du biais : en effet, il y aurait une perte de radiance dans les cas où un chemin serait arrêté trop tôt, et donc un non respect de la loi de conservation de l'énergie. Le principe de la roulette russe est utilisé pour éviter ce désagrément. La roulette russe permet la terminaison stochastique des chemins. Cela assure que notre estimateur de Monte Carlo reste non-biaisé, et donc le résultat exact pour ∞ échantillons.

Le principe est simple : après qu'un rayon ait rebondi dans la scène, la terminaison du tracé du chemin en cours va être aléatoirement décidée, selon une certaine probabilité. Cette probabilité, si elle est fixe (par exemple, 90% de chance de continuer le chemin), entraîne elle aussi un biais. Cela revient au même que d'arrêter le chemin à une profondeur arbitraire. Il faut donc que cette probabilité de poursuivre un chemin varie. Afin de réduire la variance de nos pixels, il est possible de la choisir proportionnellement à la valeur de la reflectivité. Cela va favoriser les chemins qui contribuent beaucoup à la couleur finale et, à l'inverse, les chemins dont la contribution est faible car peu de lumière est réfléchie auront une probabilité de s'arrêter plus importante. Ainsi, il est assuré que le temps de calcul ne soit pas gaspillé pour des chemins que nous savons ne contribuer que très peu au résultat final.

Pour terminer, la radiance est multipliée par le facteur de la roulette russe (1 sur la probabilité de continuer choisie) pour tenir compte de la perte d'énergie qui survient quand un chemin est terminé prématurément.

Modèles de BRDF

Ainsi, le réalisme de l'image rendue dépend maintenant principalement de la bonne définition des matériaux (et donc de leurs BRDF), domaine dans lequel de nombreux efforts ont été fournis aussi bien avant la parution de l'équation du rendu que dans les années 1990 et 2000. Ce n'est néanmoins pas directement lié au sujet de ce mémoire et nous nous contenterons donc de ne donner ici qu'une référence faisant la revue des BRDF les plus pertinentes [MU12]. À noter l'arrivée de la BRDF Disney la même année, en 2012 (voir le cours SIGGRAPH de 2012 et les notes associées *Physically Based Shading at Disney* de Brent Burley).

3.3 Colorimétrie

Grâce à l'algorithme de path tracing présenté précédemment nous pouvons maintenant obtenir une mesure approchée de la radiance qui arrive en chaque pixel de notre image. C'est cette radiance qui dans le monde réel donne l'information à notre cerveau de la brillance d'un objet et de sa couleur.

Ce qui est ici appelé radiance, est en fait le résultat de l'intégration de la radiance spectrale (radiance de la surface considérée par rapport à la longueur d'onde) sur les différentes longueurs d'onde du spectre visible de la lumière. C'est donc la distribution de cette radiance spectrale qui est en partie responsable de notre perception de la couleur.

3.3.1 Distribution Spectrale de Puissance

Souvent trouvée en anglais, *Spectral Power Distribution*, une densité spectrale de puissance correspond au flux énergétique par unité de fréquence ou de longueur d'onde. C'est à dire à une puissance par fréquence ou longueur d'onde. Les illuminants ont des DSP différentes, et sont bien sûr très importants dans le résultat final d'un rendu.

Une couleur peut être définie au moyen d'une DSP, représentant donc la puissance pour chaque longueur d'onde du spectre visible. Un spectre est continu dans la réalité, mais nous ne pouvons en avoir ici qu'un échantillonnage discret (à 1nm, 5nm, 10nm,...). La précision de cette discrétisation a un impact sur notre perception de la couleur. Habituellement, un observateur n'est pas capable de discerner de différence avec un pas inférieur ou égal à 5 nanomètres.

En l'état actuel des choses, aucune mention de la longueur d'onde de la lumière n'est faite dans l'équation du rendu (Équation 3.1). Il faut donc l'étendre pour permettre la prise en compte de la lumière en temps que spectre, et non plus se contenter de couleurs RGB définies arbitrairement et sans aucune réalité physique.

$$L_o(p, \omega_o, \lambda) = L_e(p, \omega_o, \lambda) + \int_{\Omega} f_r(p, \omega_o, \omega_i, \lambda) L_i(p, \omega_i, \lambda) (\omega_i \cdot \vec{n}) d\omega_i \quad (3.3)$$

Avec λ une longueur d'onde particulière fixée de la lumière. Cela implique que pour obtenir une représentation spectrale de notre couleur, il nous faudra en plus intégrer sur les longueurs d'onde.

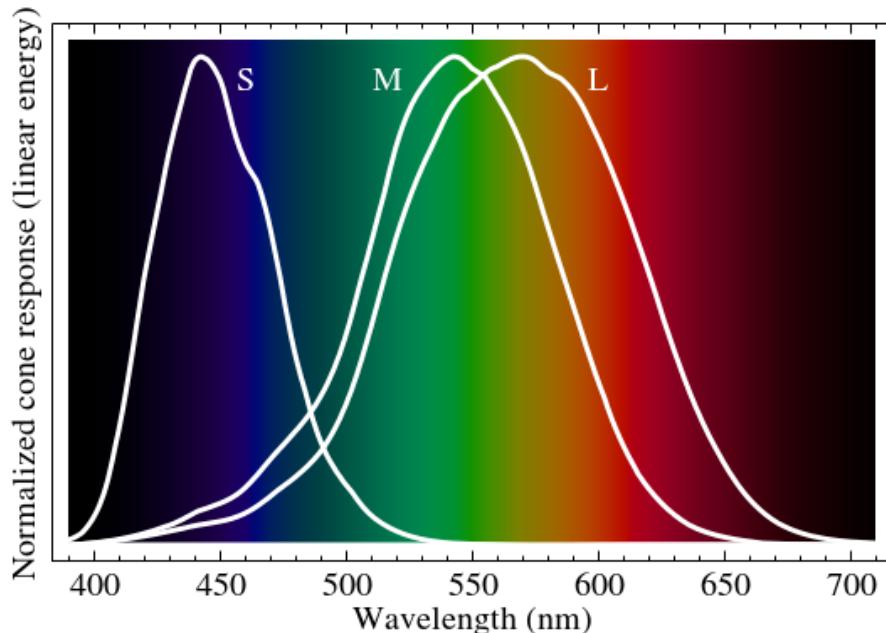


FIGURE 3.4 – Réponse normalisée des cônes humains aux stimulus mono-chromatiques du spectre visible

3.3.2 Perception humaine de la couleur

Notre rétine est tapissée, entre autres, de trois types de cellules dont les différentes sensibilités spectrales sont responsables de notre vision tri-chromatique diurne : les cônes B (en anglais S pour *small*), V (en anglais M pour *medium*) et R (en anglais L pour *long*). Chaque type est sensible à une certaine partie du spectre visible (voir Figure 3.4).

3.3.3 Du spectre à la couleur du pixel

La Commission Internationale de l’Éclairage (CIE) a défini au cours des années différents espaces de couleurs, modèles permettant une description des couleurs en accord avec la vision tri-chromatique humaine. En 1931, elle définit l'espace CIE XYZ, qui permet à partir d'un spectre d'obtenir trois valeurs X, Y et Z. Ces valeurs, le tri-stimulus XYZ, sont analogues aux réponses des cônes R, V et B. À partir des tables de Smith et Guild [TJ31], des fonctions de correspondance des couleurs ont été définies par la CIE permettant la conversion d'un spectre en tri-stimulus XYZ. Il est ensuite possible de convertir le tri-stimulus en espace de couleur RGB et ainsi obtenir une couleur affichable par un moniteur. Pour une explication plus détaillée voir les chapitres 5.2.1 et 5.2.2 de PBRT [PH10].

Cette couleur peut encore subir différents traitements. Dans le cas qui nous intéresse ici nous présentons la mise en correspondance tonale (*tone mapping*) et la correction gamma.

Mise en correspondance tonale

C'est une étape nécessaire car il est parfois difficile, voire impossible, de reproduire la luminosité réelle d'un environnement sur une image. Les contrastes qui sont perceptibles par un œil humain vont bien au-delà des capacités de nos moniteurs. C'est là qu'intervient le *tone mapping*. Son but est de réduire les fortes variations d'intensité lumineuse dans l'image pour en préserver les détails. L'article de Erik Reinhard et al. adresse ce problème [Rei+02].

Correction gamma

La vision humaine ne perçoit pas la luminosité de manière linéaire. Il est plus aisément pour l'œil humain de percevoir des différences entre deux tons sombres proches qu'entre deux tons clairs proches. Une image encodée sans correction gamma pourrait donc poser problème puisque sans correction, autant d'importance est accordée aux zones sombres qu'aux claires. La correction gamma est donc un moyen d'encoder les couleurs d'une image suivant une fonction puissance proche de notre façon de percevoir la luminances (et donc les contrastes).

3.4 Fluorescence

Dans cette section nous présentons les concepts théoriques liés à la fluorescence qui sont indispensables à la compréhension du problème posé. Elle n'a pas pour but d'être exhaustive et ne détaillera pas la physique sur laquelle les mécanismes de la fluorescence reposent.

La fluorescence est un phénomène physique caractérisant les matériaux et substances qui absorbent la lumière à une longueur d'onde donnée pour la réémettre à une longueur d'onde (presque toujours) plus élevée. La première observation de ce phénomène est faite en 1560 par Bernardino de Sahagún, mais le terme de fluorescence n'apparaît que bien plus tard, en 1852, introduit dans un livre de George Gabriel Stokes [Sto52].

3.4.1 Déplacement de Stokes

Le déplacement de Stokes, appliqué à la fluorescence, représente la différence (souvent une longueur d'onde, en nanomètres), entre les positions des deux pics des spectres d'absorption et d'émission d'un matériau (Figure 3.5). Quand un système absorbe un photon d'une longueur d'onde λ_a , il entre dans un état d'excitation et va chercher à revenir à l'état d'équilibre. Plusieurs solutions sont possibles, l'émission d'un photon, la dissipation de l'énergie sous forme de chaleur, etc. Il faut à un fluorophore³ un temps de l'ordre de 10^{-8} secondes pour retrouver son état électronique d'équilibre après excitation. Il va donc émettre un photon de plus faible énergie à l'issue de ce temps (d'une longueur d'onde $\lambda_e > \lambda_a$ ⁴). L'énergie du photon émis est plus faible car une partie de celle-ci se

3. Composé chimique fluorescent. Il est en général composé de plusieurs molécules et est capable de réémettre de la lumière après excitation.

4. Car $\lambda = c/v$, avec c la vitesse de la lumière et v la fréquence de l'onde.

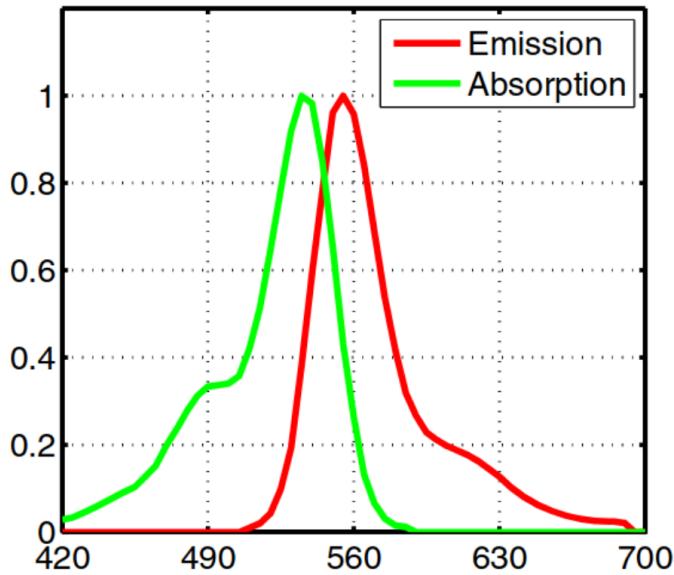


FIGURE 3.5 – Déplacement de Stokes entre les spectres d'absorption et d'émission d'un fluorophore (ordonnées : intensité; abscisses : longueur d'onde)

sera propagée aux molécules proches par relaxation vibrationnelle ou bien dissipée sous forme de chaleur. C'est ce phénomène qui est à l'origine du déplacement de Stokes.

3.4.2 Modèle de Glassner

Le problème du rendu de la fluorescence n'est pas récent. Il a été théorisé et modélisé il y a près d'un quart de siècle par Glassner [SG94] mais était alors trop complexe pour être intégré dans les moteurs de rendu de l'époque. En effet, prendre en compte cet effet dans la génération d'une image de synthèse est très consommateur en ressources : cela oblige à utiliser un modèle tenant compte de la longueur d'onde de la lumière, et donc à intégrer sur les longueurs d'onde pour résoudre l'équation du rendu, augmentant ainsi considérablement le temps de rendu et la complexité du système.

On peut donc étendre l'équation du rendu spectral (Équation 3.3) pour prendre en compte la fluorescence d'une surface comme suit :

$$L_o(p, \omega_o, \lambda) = L_e(p, \omega_o, \lambda) + \int_{\Omega} f_r(p, \omega_o, \omega_i, \lambda) \int_{\Lambda} f_f(p, \lambda' \rightarrow \lambda) L_i(p, \omega_i, \lambda) d\lambda' (\omega_i \cdot \vec{n}) d\omega_i \quad (3.4)$$

Avec :

- Λ le domaine des longueurs d'onde du spectre visible de la lumière ;
- $f_f(p, \lambda' \rightarrow \lambda)$, facteur que Glassner appelle l'efficacité de la fluorescence. Il représente la proportion d'énergie transférée entre les longueurs d'onde λ' et λ . En d'autres termes, la quantité d'énergie absorbée par les longueurs d'onde λ' du domaine Λ sur lequel a lieu l'intégration et réémise à la longueur d'onde λ .

À noter, une propriété importante des surfaces fluorescentes mise en évidence par la communauté de l'informatique graphique, et non la communauté de la physique qui

focalise ses observations sur le niveau moléculaire du phénomène de la fluorescence plutôt que sur son aspect macroscopique (visuel) : l'énergie qui est irradiée d'une surface par fluorescence rayonne de manière diffuse (uniformément dans toutes les directions). Ainsi, la composante spéculaire ne subira jamais un décalage en longueur d'onde dû à la fluorescence car la lumière n'est pas absorbée et donc pas affectée par les molécules responsables de la perte en énergie des photons [Wil+06].

Grâce à cette propriété de nouvelles BRDF analytiques ont pu être modélisées pour représenter toujours plus fidèlement les matériaux fluorescents.

Partie 4

Niixtracer, un moteur de rendu physiquement réaliste

Cette quatrième partie présente le système développé au cours de ce stage, le moteur de rendu Niixtracer. Ses fonctionnalités les plus pertinentes sont détaillées, en accord avec l'état de l'art établi précédemment.

J'ai écrit le moteur en totale autonomie et suis l'auteur de toutes ses fonctionnalités à ce jour. Pour le développement j'ai utilisé la chaîne d'outils suivante : environnement UNIX, compilation avec CMake et g++, l'IDE QtCreator. J'ai utilisé git pour la gestion du code. Il est disponible sur GitHub et bien entendu ouvert aux contributions.

4.1 Fonctionnalités

Cette section dresse un constat des fonctionnalités implémentées dans le moteur de rendu développé pour le laboratoire.

4.1.1 Fichiers de scène

Un fichier de scène est une façon de représenter textuellement une scène virtuelle 3D. C'est donc un moyen de décrire une scène virtuelle 3D qu'un être humain peut aisément appréhender. C'est ce fichier de scène que le moteur de rendu est chargé de parcourir et analyser. Il va, à partir des informations collectées dans le fichier de scène, calculer en sortie l'image de synthèse correspondante.

Qt XML

La partie du moteur chargée de lire les fichiers de scène est la seule du système à reposer sur une bibliothèque externe. La bibliothèque en question est la bibliothèque utilitaire Qt, et plus particulièrement le module Qt XML. Ayant déjà été amené à utiliser cette bibliothèque pour lire et écrire des fichiers de scène c'est un choix qui m'a paru évident.

Le moteur va donc lire le fichier de scène en XML et, s'il est bien formaté, instancier les structures de données nécessaires pour pouvoir générer l'image.

Vous trouverez en annexe une explication plus détaillée du format utilisé pour définir des scènes lisibles par Niixtracer (voir Annexe A).

Formes géométriques, plans et sphères

Les formes géométriques prisent en charge par Niixtracer sont très simples et ne sont qu'au nombre de deux : les plans et les sphères. Néanmoins, l'architecture telle qu'elle est pensée permet aisément d'implanter d'autres formes : cylindres, parallélépipèdes, faces triangulaires d'un modèle, etc.

Le but du moteur n'étant pas de rendre des scènes très diversifiées, sphères et plans suffisent : il n'est pas du tout primordial pour observer les effets de fluorescence qui nous intéressent dans le cadre de ce mémoire d'avoir plus de formes.

4.1.2 Ray tracing

Le moteur de rendu développé permet de rendre une scène en utilisant un algorithme de ray tracing récursif classique, tel que présenté par Whitted [Whi80]. Il utilise la géométrie optique pour prendre en compte réflexions, réfractions et ombres. Pour un exemple des effets implémentés, voir la figure 4.1. D'autres rendus sont disponibles en annexes B.1 et B.2.

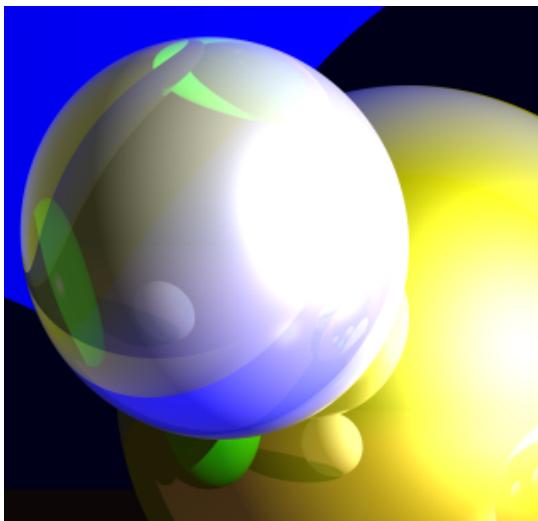
Réflexions et réfractions

Pour connaître la proportion de lumière réfléchie par rapport à la proportion de lumière transmise, il faut résoudre les équations de Fresnel. Elles nous renseignent sur le comportement d'une onde à l'interface entre deux milieux, ici la surface d'un objet de notre scène. Pour des résultats plus rapides à calculer il est possible d'utiliser l'approximation de Schlick [Sch94]. Le comportement de l'onde à l'interface dépend (principalement) des indices de réfraction des milieux d'entrée et de sortie et de l'angle d'incidence de l'onde avec l'interface.

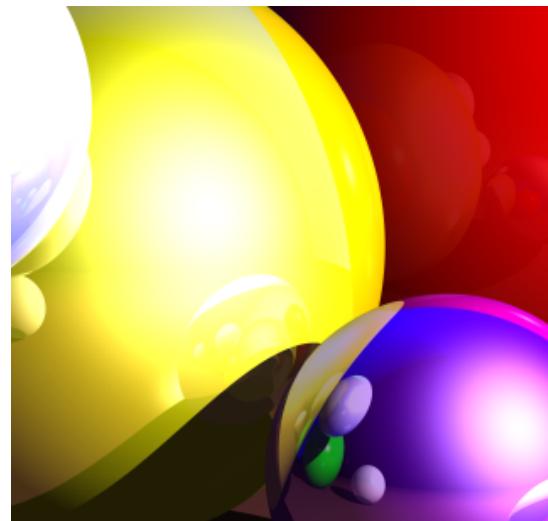
Ombres

À chaque intersection d'un rayon avec un objet, un rayon d'ombre est lancé (*shadow ray*) avec pour origine le point de la surface et pour direction la source de lumière pour vérifier que celle-ci est bien visible (et donc que le point de la surface intersectée n'est pas dans l'ombre). Avec cet algorithme les sources de lumière utilisées sont ponctuelles (*point lights*) et ont donc une position dans l'espace que l'on peut utiliser comme cible pour lancer nos *shadow rays*.

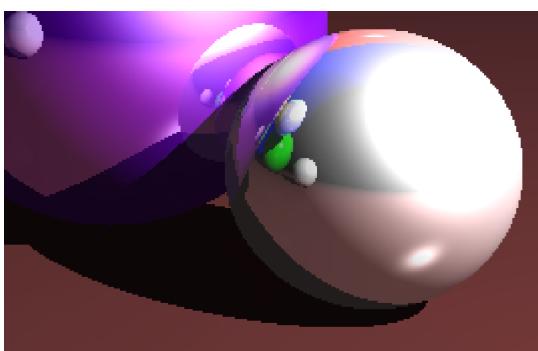
À noter qu'il faut décaler l'origine des rayons récursifs d'un epsilon fixé vers la surface pour éviter les problèmes d'auto-ombrage (*self-shadowing*, en anglais) ou d'auto-intersection (*self-intersection*) dans le cas des réflexions.



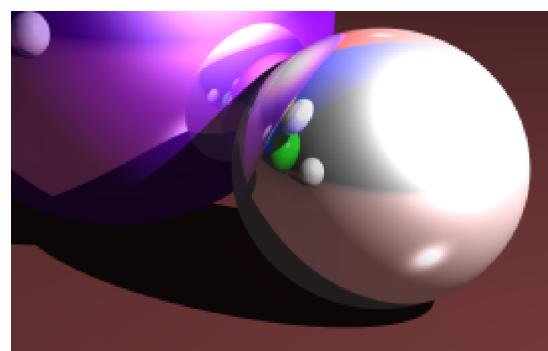
(a) Sphère qui réfracte la lumière ($\text{IoR} = 2.4$)



(b) Multiples réflexions car rebonds max > 1



(c) Pas d'antialiasing (crénelage sur les contours des sphères et des ombres)



(d) Idem que 4.1c mais avec antialiasing (pixels subdivisés 4 fois)

FIGURE 4.1 – Détails d'une scène rendue au moyen de l'algorithme de ray tracing récursif. La sphère verte qui se reflète est hors champs, sur la gauche. De même pour la source qui est en hauteur sur la droite.

Nombre de rebonds d'un rayon

Comme dit précédemment, l'algorithme de ray tracing étant récursif, il faut choisir à quel moment arrêter les rebonds du rayon dans la scène. Pour cela un nombre de rebonds maximal est défini. Ce n'est bien sûr pas la meilleure méthode si l'objectif visé est celui du photoréalisme (voir la section 3.2.2. qui traite du path tracing pour cela).

Antialiasing

Ce mode de rendu permet aussi de définir un niveau d'antialiasing (voir Figure 4.1c et 4.1d). La méthode utilisée est somme toute assez simple : chaque pixel est subdivisé en sous-pixels plus petits et un rayon est lancé au travers de chacun d'entre eux. Ensuite la moyenne de leurs contributions est faite. De cette façon les contours des sphères paraissent lisses et le phénomène de crénelage est évité.

4.1.3 Path tracing

Un autre mode mis à disposition par le moteur pour rendre une image a recours à un algorithme de path tracing, tel que décrit par Kajiya [Kaj86]. Cette fois, plus question d'optique géométrique, les différents matériaux et leurs interactions avec la lumière étant gérés au moyen de *Bidirectional Reflectance Distribution Functions*.

Bidirectional Reflectance Distribution Functions

Les BRDF implémentées à ce jour sont au nombre de quatre. Ce sont toutes des BRDF analytiques, issues de modèles mathématiques plutôt que d'expériences comme cela peut parfois être le cas. On a donc :

- Modèle diffus parfait;
- Modèle spéculaire parfait (miroir);
- Modèle « brillant » (probabilité de spéculaire parfait + diffus);
- Modèle réfringent¹ parfait.

Ces modèles peuvent être répartis entre deux types de matériaux : les métaux et les diélectriques.

Les métaux ont une rugosité, comprise entre 0 et 1. Ce paramètre permet d'obtenir des surfaces plus ou moins brillantes. À zéro, tous les rayons sont réfléchis dans la direction spéculaire idéale (cas particulier du miroir). Mis à 1, il permet de modéliser une surface idéalement diffuse. Une telle surface n'existe pas dans la réalité mais certains matériaux s'en rapprochent beaucoup (l'argile et la craie par exemple).

Les diélectriques quant à eux, sont des milieux isolants. Ils peuvent se trouver sous différents états (solide, liquide, gazeux). Dans le cas des solides, seul cas qui nous intéresse pour l'instant, certains diélectriques ont la propriété intéressante d'être transparents (notamment le verre). Ceux-ci ont donc un indice de réfraction propre à chacun².

Un exemple des matériaux implémentés dérivés de ces deux types est visible sur la figure 4.2.

Échantillonnage explicite des sources

Le moteur dispose d'une fonctionnalité permettant l'échantillonnage explicite des sources (*Next Event Estimation, NEE*, en anglais, ou bien encore *Explicit Light Sampling*). Cette technique permet d'augmenter considérablement la vitesse de convergence de notre algorithme.

Le principe est relativement simple : plus l'aire d'une source est faible, moins la probabilité qu'un chemin l'atteigne est grande. Il existe donc des scènes pour lesquelles de nombreux chemins auront une contribution nulle, autrement dit, du temps de calcul perdu en vain. Une solution à ce problème est d'aller échantillonner la lumière explicitement, à la manière d'un *shadow ray*. Pour chaque point d'une surface diffuse qui serait frappée par un rayon, un rayon est directement lancé vers la source. Si elle est visible, la

1. Terme français pour l'anglais *refractive*. Larousse : « Qui produit la réfraction. »

2. Liste des indices de réfraction de nombreux diélectriques : <https://pixelandpoly.com/ior.html>

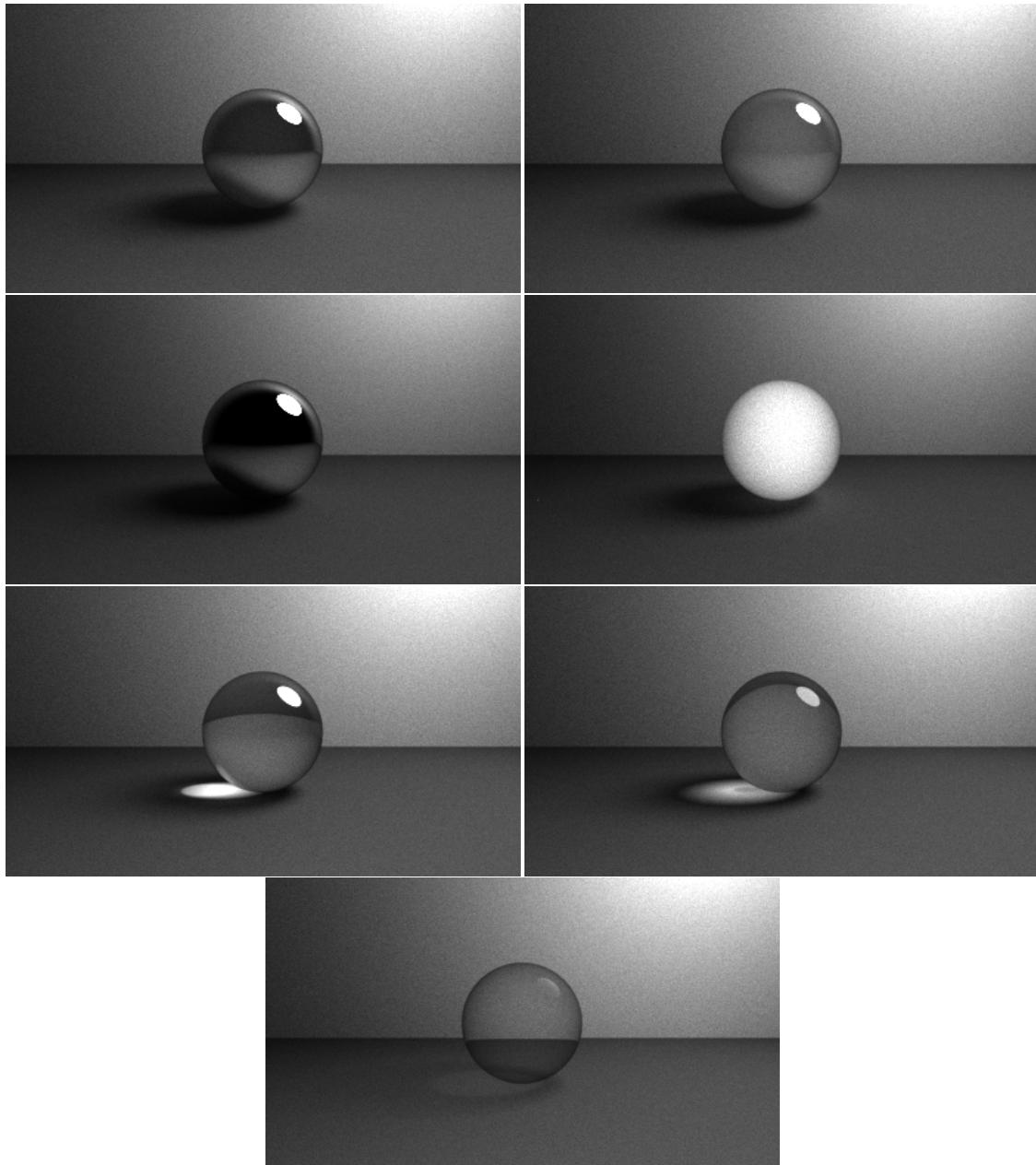


FIGURE 4.2 – Sept matériaux différents, ligne par ligne, de gauche à droite : métal (rugosité = 0.3), plastique (rugosité = 0.7), miroir (rugosité = 0), matte (rugosité = 1), diamant ($\text{IoR} = 2.4$), verre ($\text{IoR} = 1.5$), transparent ($\text{IoR} = 1.02$)

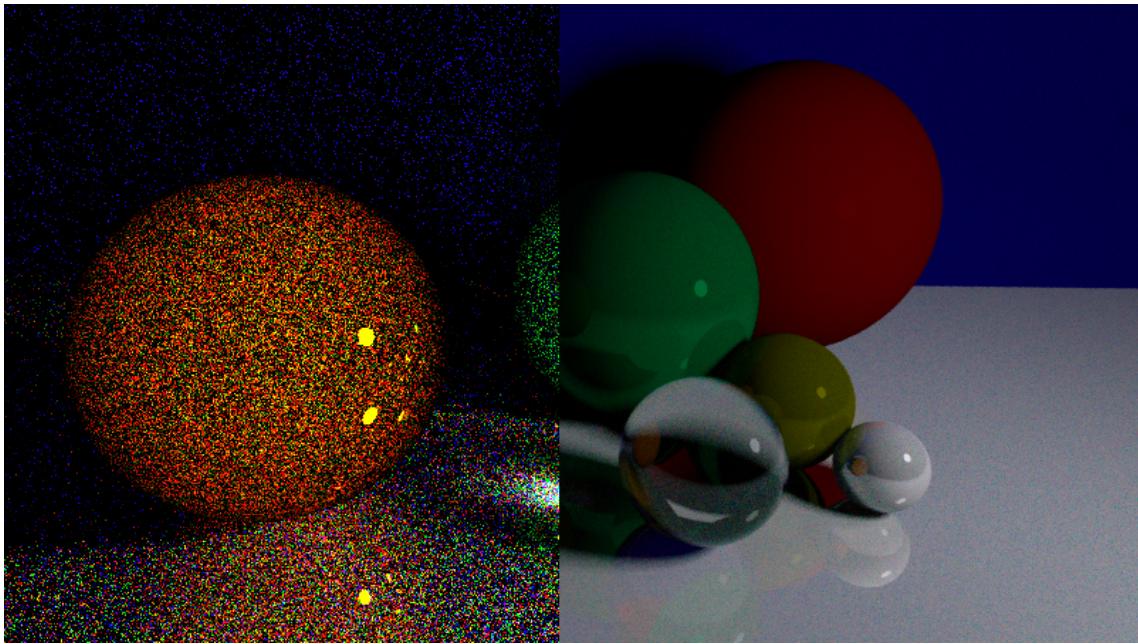


FIGURE 4.3 – Exemple d’une même scène, rendue sans (à gauche) et avec (à droite) échantillonnage explicite des sources. À noter la présence de la caustique sur l’image du path tracing classique. Nombre d’échantillons par pixel : 1024

contribution de la lumière est directement ajoutée au chemin. Pour que cela fonctionne correctement et que le montant total d’énergie distribué dans la scène reste constant (respect de la conservation de l’énergie, encore une fois), la contribution de la source n’est plus ajoutée quand un chemin atteint celle-ci indirectement. Il est inutile de l’ajouter à ce moment là puisqu’elle a déjà été prise en compte tout au long du chemin, à chaque intersection diffuse.

Cet échantillonnage explicite des sources permet de séparer illumination directe et indirecte. Cela peut s’avérer utile, par exemple aux artistes dans l’industrie du cinéma, afin de leur donner un meilleur contrôle sur le rendu d’une scène, par exemple en accentuant les contributions de l’illumination directe, ou en privilégiant les contributions indirectes d’une source en particulier, à un nombre de rebonds donné, etc.

Dans notre cas, cette séparation entre illumination directe et indirecte n’est pas primordiale. Néanmoins l’échantillonnage explicite des sources est tout de même appréciable car il permet de réduire considérablement la variance à nombre d’échantillons égal, et donc le nombre d’échantillons nécessaires pour obtenir la qualité de rendu souhaitée (voir Figure 4.3). En revanche, il ne permet pas en l’état le rendu de caustiques. Il existe un article traitant de ce problème et proposant une solution pour reconnecter les surfaces pour lesquelles la source serait occultée par d’autres objets réfringents [HDF15]. À ce jour, cette amélioration ne fait pas partie de Niixtracer, et encore une fois n’est pas prioritaire.

4.1.4 Spectral path tracing

Nous en arrivons maintenant au *spectral path tracing*, extension de l’algorithme original. Au lieu d’utiliser un espace de couleur comme l’espace RGB pour représenter la couleur de nos objets, ce qui n’a pas de cohérence physique réelle, le *spectral path tracing*

repose sur les Distributions Spectrales de Puissance pour représenter les couleurs.

Distributions Spectrales de Puissance

Comme présenté dans l'état de l'art, une DSP correspond au flux énergétique (une puissance) par fréquence ou par longueur d'onde. C'est ce qui est utilisé dans le moteur pour définir la proportion de la radiance spectrale réfléchie par une surface à une longueur d'onde donnée. Quand on utilise le *spectral path tracing* chaque objet possède donc *a minima* un spectre de réflexion. On trouve deux types très simples de DSP implémentées dans le moteur.

Distribution constante Cette distribution, comme son nom l'indique a un flux énergétique invariant selon la longueur d'onde. De cette façon, si une surface réfléchie la lumière de manière égale à toutes les longueurs d'onde du spectre visible, elle apparaîtra blanche.

Distribution normale Cette distribution suit une loi normale. Elle permet d'avoir une DSP centrée sur une longueur d'onde donnée, et donc privilégier une certaine couleur.

$$f(\lambda | \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\lambda-\mu}{\sigma}\right)^2}$$

Avec,

- λ la longueur d'onde considérée;
- σ la déviation standard, σ^2 la variance;
- μ l'espérance mathématique.

Illuminants

Les illuminants standards, définis par la CIE, possèdent différentes DSP. Ces illuminants possèdent tous une température de couleur proximale (*Correlated Color Temperature, CCT*, en anglais) propre.

Loi de Planck Cette température de couleur proximale³ permet d'obtenir grâce à la loi de Planck des DSP proches de celles des illuminants standards définis par la CIE (voir Figure 4.4). Cette loi définit la radiance spectrale par longueur d'onde émise par un corps noir⁴ en fonction de la seule température (en Kelvin) de ce corps.

3. "The correlated color temperature (CCT) is the temperature of the Planckian radiator whose perceived color most closely resembles that of a given stimulus at the same brightness and under specified viewing conditions" - CIE/IEC 17.4 :1987, International Lighting Vocabulary

4. Wikipédia : « Le corps noir est un objet idéal qui absorberait parfaitement toute l'énergie électromagnétique qu'il reçoit, sans en réfléchir ni en transmettre. Il n'est fait aucune autre hypothèse sur la nature de l'objet. Sous l'effet de l'agitation thermique, le corps noir émet un rayonnement électromagnétique. À l'équilibre thermique, émission et absorption s'équilibreront et le rayonnement effectivement émis ne dépend que de la température (rayonnement thermique) »

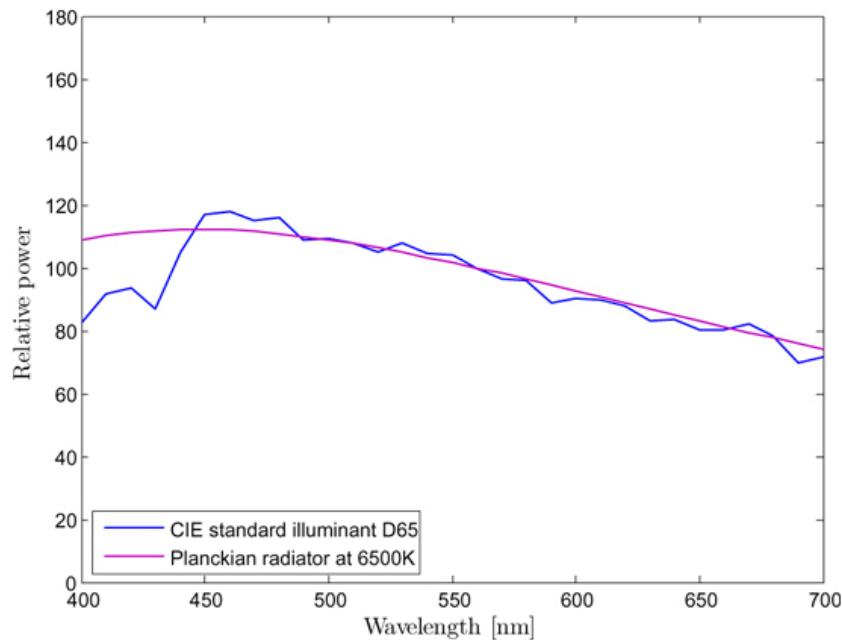


FIGURE 4.4 – DSP de l'illuminant D65 et d'un corps noir de même température. Figure issue de www.image-engineering.de

La loi de Planck s'exprime ainsi :

$$L_{\Omega,\lambda}^{\circ}(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{hc/(\lambda kT)} - 1}$$

Avec,

- λ la longueur d'onde considérée ;
- h la constante de Planck ($= 6.626070040(81) \cdot 10^{-34} J.s^{-1}$) ;
- c la vitesse de la lumière ($= 299792458 m.s^{-1}$) ;
- k la constante de Boltzmann ($= 1.38064852 e - 23 J.K^{-1}$) ;
- T la température de la surface (en Kelvin K).

Il est clair que la seule variable dans cette formule est la température de la surface du corps noir pour une longueur d'onde fixée.

Voici les différents illuminants implémentés dans le moteur à ce jour :

Illuminant A Cet illuminant a été conçu pour correspondre à un éclairage d'intérieur, typiquement avec une ampoule à filament de tungstène. C'est pourquoi il donne une teinte jaune aux scènes dans lesquelles il est utilisé (voir Figure 4.5).

Illuminants D La série des illuminants D est une tentative pour caractériser la lumière du jour. Elle comporte quatre illuminants :

- L'illuminant D50, qui représente la lumière du jour à l'horizon ;
- L'illuminant D55, pour la lumière du jour en milieu de matinée ou d'après-midi ;
- L'illuminant D65, le plus répandu. Il correspond à la lumière du jour à midi ;

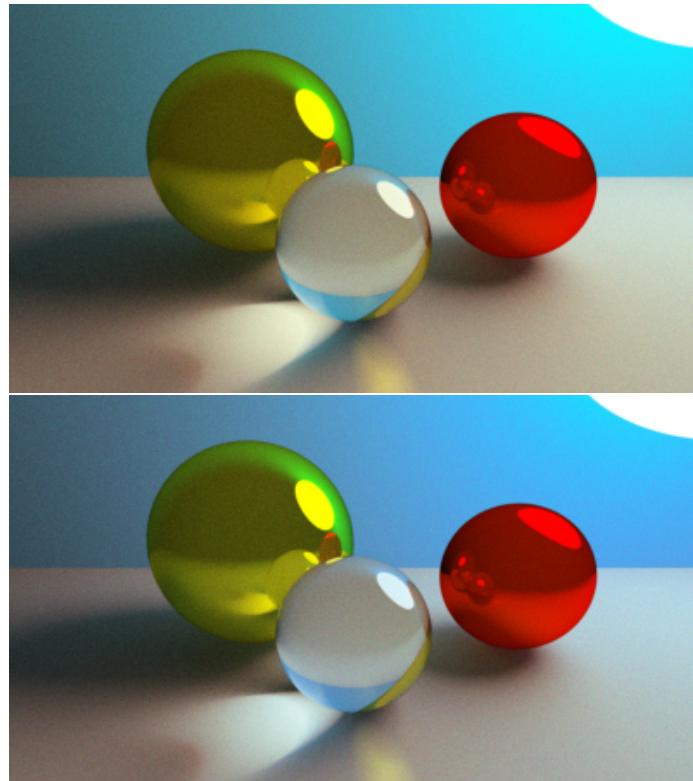


FIGURE 4.5 – Illuminants A (en haut) et D65 (en bas). La différence de couleur est surtout visible sur la caustique et le plan bleu. Nombre d'échantillons par pixel : 1024

- Et enfin l'illuminant D75, qui correspond lui aussi à la lumière du jour à midi, mais pour un ciel plus proche du pôle Nord (lumière plus bleue, donc).

Lumière noire Ce qui est communément appelé lumière noire est une source de lumière qui émet principalement des ultra-violets. La DSP d'une telle source comporte un pic très important autour de 370nm (en général, mais cela dépend du composé chimique utilisé) et un second pic plus faible autour de 400 nanomètres, ce qui va produire un rayonnement donnant à la source cette teinte violette-magenta caractéristique des lumières noires.

4.1.5 Fluorescence

Pour implémenter l'effet de fluorescence il a fallu concevoir de nouveaux matériaux qui possèdent des spectres d'émission et d'absorption en plus de leur BRDF.

Distribution de Cauchy asymétrique

Afin d'obtenir des spectres d'émission et d'absorption qui soient fidèles au comportement réel de surfaces fluorescentes, une distribution de Cauchy asymétrique est utilisée (voir Figure 4.6), suivant le modèle à trois paramètres proposé par Yinqiang Zheng et al. [ZSS14]. Il repose sur la formule suivante :

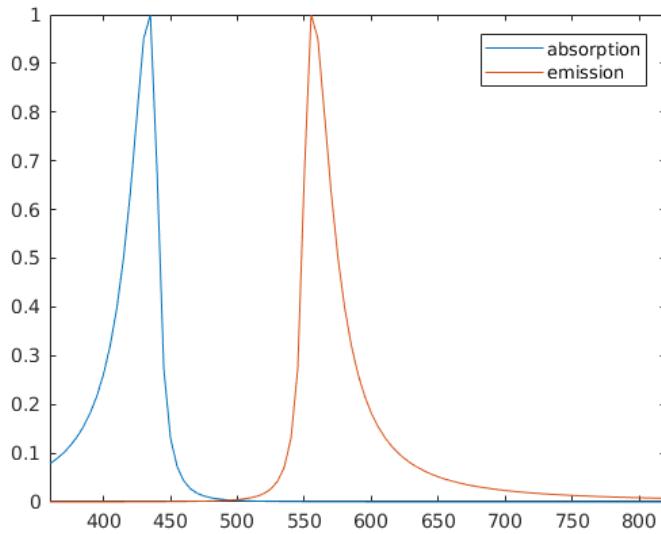


FIGURE 4.6 – Spectres d’absorption (bleu) et d’émission (rouge) (matériau numéro 50 de la base de donnée [McN+06])

$$f(\lambda | \lambda_0, \gamma, \omega) = \frac{\gamma}{\pi[\gamma^2 + (\lambda - \lambda_0)^2]} \left\{ \frac{1}{\pi} \arctan\left[\frac{\omega(\lambda - \lambda_0)}{\gamma}\right] + \frac{1}{2} \right\}$$

Avec,

- λ_0 le pic de longueur d’onde du spectre;
- γ la largeur de la distribution;
- ω le coefficient d’asymétrie de la distribution.

Ce modèle permet de modéliser aussi bien un spectre d’émission qu’un spectre d’absorption, comme prouvé dans l’article. Le coefficient d’asymétrie doit être positif pour le spectre d’émission et négatif pour celui d’absorption ($\omega_a = -\omega_e$).

Un rendu d’une scène sans et avec fluorescence est visible en figure 4.7. Dans cette scène, les deux sphères colorées sont fluorescentes. La sphère bleue à gauche absorbe les ultra-violets et réemet autour des longueurs d’onde 540 nm (vert), la sphère jaune, elle, absorbe dans les longueurs d’onde bleues et réemet autour de 680 nm (rouge). La scène fluorescente gagne en radiance totale, car les ultra-violets absorbés sont réémis dans le spectre visible, pouvant donner l’impression que la scène est plus lumineuse. À noter que conformément aux attentes, la composante spéculaire n’est pas affectée par la fluorescence. Plus de rendus et des zooms sont consultables en annexe E.

4.2 Résultats

4.2.1 Nombre d’échantillons

Le nombre d’échantillons par pixel utilisé pour le rendu d’une image est directement lié à la variance de l’image obtenue par rapport au résultat exact attendu. Comme nous l’avons vu, le taux de convergence des méthodes de Monte Carlo est très lent ($O(1/\sqrt{n})$).

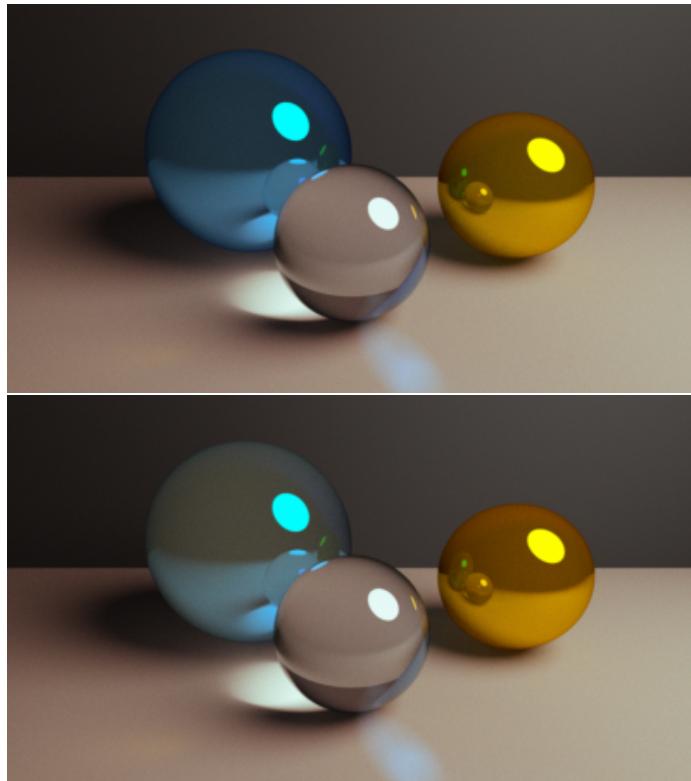


FIGURE 4.7 – Comparaison de la même scène sans (en haut) et avec (en bas) fluorescence

Il faut donc, pour diviser l'erreur par deux, quadrupler le nombre d'échantillons (voir Figure 4.8). On peut donc fixer un seuil d'erreur au dessous duquel on considère le rendu comme correct et arrêter l'algorithme une fois ce seuil atteint.

4.2.2 Performances

Les performances n'étaient pas du tout au cœur de nos préoccupations au cours du développement. En effet, à terme, les images qui seront rendues avec le moteur par l'équipe seront de petite taille et les scènes très simples pour pouvoir éprouver des mesures effectuées en laboratoire au moyen d'une caméra spectrale, par exemple. Néanmoins, les performances du moteur restent acceptables et un rendu peut s'effectuer dans un temps humainement tolérable : de l'ordre de l'heure pour un rendu avec une racine de l'erreur quadratique moyenne autour de 3. Bien sûr, la convergence dépend de la complexité de la scène rendue et celle-ci n'est donc jamais garantie, mais les résultats présentés et les temps de rendu ont été jugés convenables compte-tenu des spécificités d'utilisation sus-mentionnés.

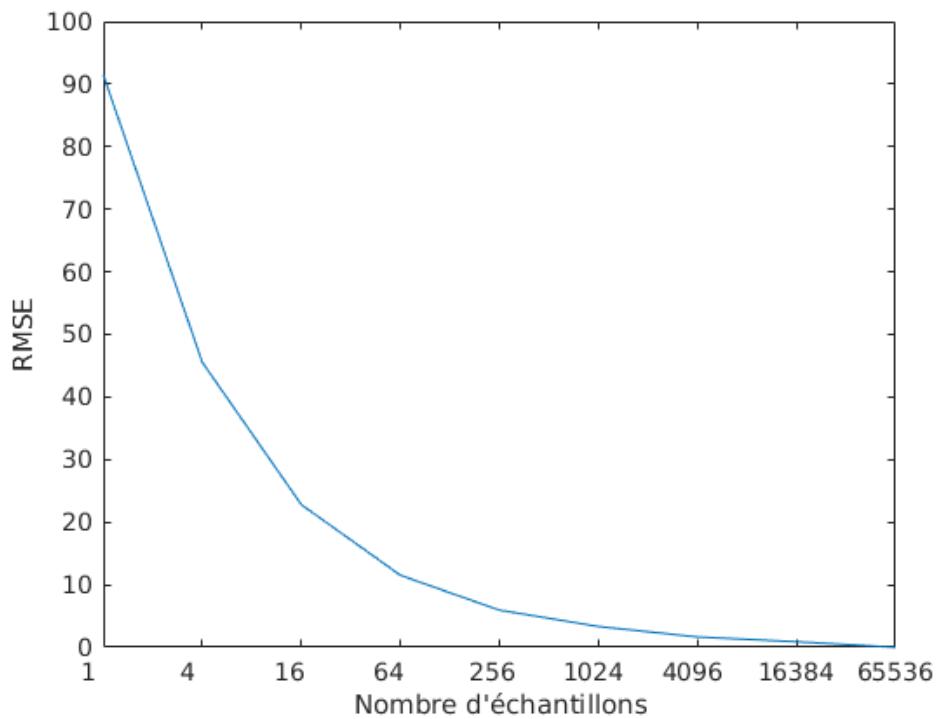
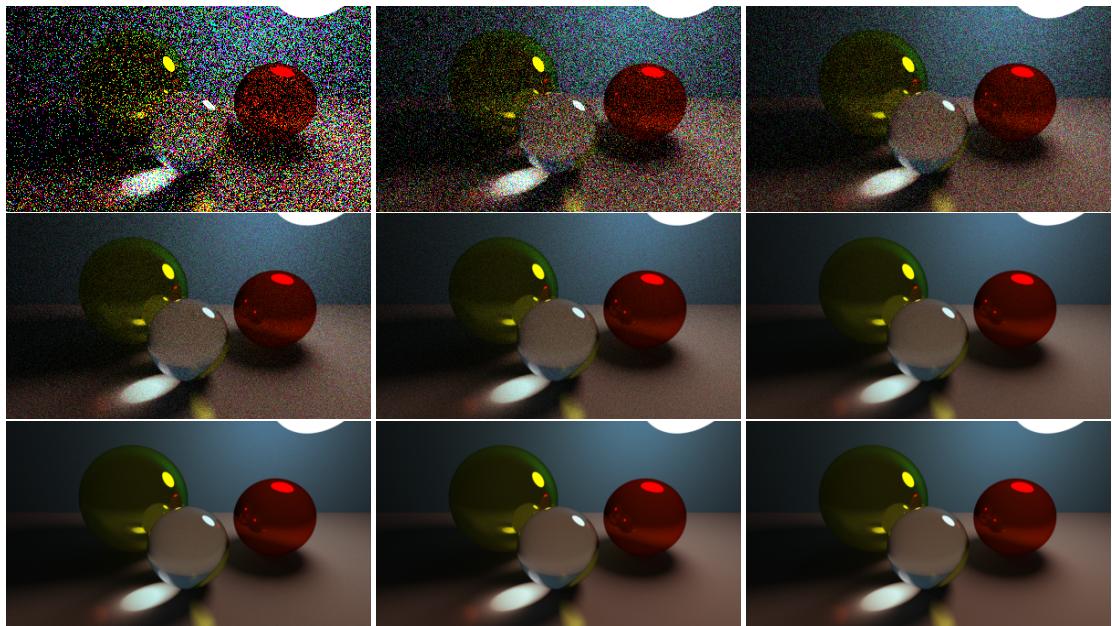


FIGURE 4.8 – Rendu à différents nombres d'échantillons d'une scène au moyen de l'algorithme de spectral path tracing sans échantillonnage explicite des sources et graphe de la racine de l'erreur quadratique moyenne par rapport au nombre d'échantillons pour cette scène.

Partie 5

Travaux futurs

Cette partie adresse les finalités du travail effectué au cours des six mois passés au NII, en quoi sera-t-il utile pour le laboratoire et comment il pourra évoluer par la suite.

5.1 Rendu de liquides fluorescents

Un des centres d'intérêt actuel de l'équipe, et plus spécifiquement du professeur Imari Sato, est l'étude de liquides fluorescents. Des mesures sont actuellement en cours pour mesurer les BRDF de différents liquides possédant une composante fluorescente. Parmi eux, on retrouve assez étonnamment des liquides assez communs, tels le miel ou le whisky. C'est une des raisons pour lesquelles travailler en ce sens et approfondir les travaux existants sur le sujet est intéressant. La fluorescence est un effet, pourtant courant, très souvent ignoré malgré sa contribution non négligeable à l'éclairage d'une scène.

En accord avec cette direction, j'ai effectué des recherches sur le sujet dans la littérature et trouvé un article traitant du sujet [Abd+17]. Celui-ci n'a pas encore été publié dans un journal (seulement sur [arXiv.org](https://arxiv.org)) mais nous a semblé prometteur et pourrait faire l'objet d'une implémentation dans Niixtracer. De même, les propriétés de diffusion de nombreux milieux participatifs sont recensées dans l'article de Srinivasa G. Narasimhan et al. [Nar+06] qu'il m'a été donné d'étudier. Cette base de données est de premier intérêt pour ce travail à venir.

Une fois implémenté, le moteur pourrait être utilisé pour générer des images de synthèse physiquement plausibles de milieux participatifs fluorescents, ce qui permettrait au laboratoire de s'en servir comme référence pour éprouver leurs mesures ou bien encore pour fournir des données d'entrée nécessaires à la résolution de problèmes inverses en vision par ordinateur (suppression des inter-réflexions pour la reconnaissance de formes par exemple).

5.2 Iridescence

À la manière de la fluorescence, l'iridescence (voir Figure 5.1), ou goniochromisme, est un phénomène optique relativement fréquent dans la nature (ailes de papillons, paons, bulles, certains insectes,...) qui n'a reçu que très peu de couverture dans le domaine de



FIGURE 5.1 – Carapace d'un scarabée présentant un changement graduel de couleur dû au phénomène d'iridescence

l'informatique graphique. Il nécessite pour être rendu un moteur de rendu spectral, qui tient compte des interférences et des décalages de phase des ondes lumineuses pouvant survenir lors des phénomènes de réflexions, mais aussi de modèles de surfaces multi-couches (*multi-layered surfaces* en anglais). Ce phénomène peut aussi survenir à cause de la diffraction de la lumière par une surface, à la manière d'un prisme.

Une partie des éléments nécessaires étant déjà présente dans Niixtracer, l'iridescence pourrait faire l'objet d'un ajout intéressant.

Partie 6

Bilans technique et personnel

Dans cette sixième et dernière partie le bilan du stage est dressé, aussi bien sur le plan technique pour le laboratoire, que sur le plan personnel.

6.1 Bilan technique

6.1.1 Pour le laboratoire

À l'issue de ce stage, le National Institute of Informatics, et plus précisément l'équipe que j'ai intégrée au sein de la division Digital Media and Media Sciences, se voit dotée d'un nouvel outil : le moteur de rendu Niixtracer.

Ce moteur permettra à l'équipe de mener à bien des expériences concrètes, mais aussi faire l'objet d'extensions futures. Ma superviseure pouvant encadrer jusqu'à trois stagiaires internationaux par an dans le domaine de l'informatique graphique et de la vision par ordinateur, en plus des doctorants qu'elle dirige et assistants de recherche qu'elle emploie, je ne doute pas qu'elle utilisera cette occasion pour faire grandir le projet, notamment sur les points abordés dans la partie 5. Bien sûr, les améliorations possibles sont nombreuses et il reste beaucoup à faire, mais la base de tout ceci est suffisamment solide pour le supporter.

Aussi, je pense que ma venue dans l'équipe a permis de réintroduire auprès de ses membres l'informatique graphique et plus particulièrement le rendu et ses problématiques. En effet, c'est un domaine qu'ils avaient délaissé au profit de la vision par ordinateur et nouvellement du *machine learning*. J'ai trouvé mes collègues très intéressés et curieux, c'est pourquoi je pense pouvoir affirmer que ce travail a été bénéfique pour toutes les parties tenantes, et je ne serai pas surpris de voir certains d'entre eux essayer de faire des ponts entre les disciplines.

6.1.2 Compétences acquises

Grâce à ce stage, j'ai pu acquérir et consolider bon nombre de compétences techniques et sociales, de connaissances acquises au cours de mes études supérieures dans le domaine de l'informatique, voire même de réviser des notions théoriques plus anciennes,

comme par exemple la physique de la lumière et l'optique géométrique. Comme souligné précédemment le moteur développé est écrit en C++, un langage qu'il est, je pense, indispensable de connaître (à défaut de pouvoir le maîtriser tant la tâche est ardue) pour pouvoir prétendre à un poste d'ingénieur dans le domaine de l'informatique graphique. J'ai donc acquis des automatismes de développement et pu travailler sur un projet de taille conséquente, et ce avant mon entrée dans la vie active. C'est, de mon point de vue, tout l'intérêt des stages qui jalonnent nos parcours universitaires et en ce sens, celui-ci a totalement rempli sa mission. Aussi, chose très importante pour moi, les nombreuses présentations de mes travaux que j'ai effectuées auprès de l'équipe ont contribué à développer mon aisance et la confiance que je peux placer en mes capacités.

Étant seul à avoir contribué à Niixtracer, cela m'a permis de contrôler tous les aspects du projet moi-même et donc d'être impliqué dans toutes les différentes étapes du développement, de me poser les questions sur la problématique à laquelle répond ce travail, sa faisabilité, la conception du système en lui-même, jusqu'aux détails plus techniques concernant les choix d'implémentation.

6.2 Bilan personnel

Ce stage était somme toute une expérience très intéressante et formatrice, à mi-chemin entre travail de recherche et travail d'ingénierie. À l'issue de celle-ci j'appréhende mieux le rôle que je serai amené à remplir dans une équipe, et le travail attendu d'un ingénieur en Informatique. De plus, le bon déroulement du stage dans sa globalité et les bons rapports entretenus avec mes collègues de travail et ma superviseure me confortent dans l'idée que je saurai trouver ma place dans une équipe, quelque soit ma position future, aussi bien dans une entreprise que dans le monde académique.

Je suis également très heureux et reconnaissant d'avoir pu vivre cette expérience si spéciale au Japon. C'est un objectif que je m'étais fixé de longue date et beaucoup de mes choix en ont été affectés. Ces six mois furent loin d'être faciles : la distance avec mes proches (dix mille kilomètres à vol d'oiseau et huit heures de décalage horaire), le gigantisme de Tokyo, être amené à sortir de ma zone de confort et à me remettre en question sur une base quotidienne, la barrière de la langue, etc. Autant d'épreuves subsidiaires qui ont contribué à me faire progresser sur le plan humain. J'en ressors profondément marqué, changé, mais surtout convaincu que mes choix étaient les bons et que c'est cette vie que je veux mener dans le futur : travailler dans des environnements internationaux, entouré de personnes expertes dans leurs domaines respectifs, dont j'ai tellement à apprendre.

6.3 Perspectives d'avenir

Mes perspectives professionnelles pour l'avenir ne sont pas encore arrêtées. Bien qu'initialement très intéressé, j'ai finalement décliné l'offre de ma superviseure de poursuivre mes études en thèse au NII, principalement pour des raisons financières. Le salaire perçu par un doctorant au Japon étant clairement insuffisant pour survivre à Tokyo sans avoir à cumuler petits emplois, temps de trajet interminables, etc, je ne me sentais pas prêt à ce sacrifice. Aussi, nous avons évoqué mon possible retour au Japon pour la prochaine année fiscale (en Avril 2019), en tant qu'ingénieur de recherche.

C'est un compromis intéressant, néanmoins je souhaiterais mener une thèse dans

un futur proche, c'est pourquoi je n'ai pas encore arrêté de décision. Une solution idéale serait peut-être une collaboration entre mon université et un laboratoire étranger (le NII ou autre) pour poursuivre mon travail.

Vivre de longues périodes à l'étranger m'apprends beaucoup sur moi-même et les autres, me rends plus conscient des bons et mauvais aspects des pays où j'ai pu résider, la France y compris, mais cela a un coût important : à trop voyager le risque est de ne plus se sentir chez soi nulle part. Impossible donc de savoir quel sera le prochain pays où je serai amené à vivre et travailler, mais cela n'a pas de réelle importance : j'apprends un peu plus à chaque pas, et ce stage n'en était pas un faux.

Partie 7

Conclusion

Ce stage de recherche en laboratoire est ma première expérience de travail excédant les deux mois habituels des stages de Licence. Cette durée plus conséquente m'a permis d'explorer au préalable l'état de l'art dans le domaine du rendu physiquement réaliste et d'en acquérir une meilleure compréhension.

Cette vision d'ensemble m'a donné les clefs nécessaires pour définir un sujet de stage qui soit à la fois pertinent et formateur pour mon parcours : le rendu de matériaux fluorescents au moyen d'un algorithme de path tracing spectral. Pour explorer ce sujet j'ai développé un moteur de rendu physiquement réaliste qui permet d'effectuer des rendus spectraux et prend en compte la fluorescence.

La fluorescence est un effet visuel qui était quelque peu délaissé par la communauté de l'Informatique Graphique, pourtant il n'est à ce jour pas encore totalement couvert comme en témoignent les récents travaux d'Adrian Jarabo et Victor Arellano [JA18] et de Michal Možík et al. [MFW18].

Bien que ma contribution soit pour le laboratoire plus proche d'un travail d'ingénierie elle s'appuie sur une méthodologie propre à un travail de recherche et pourrait déboucher sur diverses extensions comme le rendu de liquides fluorescents non-homogènes.

Pour conclure, ma volonté de poursuivre en thèse dans un contexte international se trouve renforcée par cette expérience et j'espère avoir l'opportunité de continuer le travail initié ici dans le futur.

Annexes

Annexe A

Fichiers de scène

Définition des paramètres utilisés par le moteur (A.1) pour produire un rendu d'une scène au format BMP, enregistrée à la localisation PATH, de résolution XRES*YRES. L'image de synthèse générée utilisera l'algorithme MODE, avec un nombre de rebond maximum (sauf dans les cas où la roulette russe est utilisée) égal à DEPTH. SAMPLES échantillons seront tirés par pixel pour rendre l'image. Dans le cas d'un rendu spectral, l'illuminant ILLUM doit être renseigné.

Code A.1 – Paramètres de l'algorithme utilisé et de l'image générée

```
<scene>      <!-- Cette balise encadre l'ensemble du fichier -->
  <settings>
    <xres>XRES</xres>
    <yres>YRES</yres>
    <path>PATH</path>
    <mode>MODE</mode>
    <depth>DEPTH</depth>
    <sampl>SAMPLES</sampl>
    <illum>ILLUM</illum>
  </settings>
  ...
</scene>
```

Code A.2 – Définition de la caméra et des sources ponctuelles (ray tracing) de la scène

```
...
<camera>
  <position>CPOSX, CPOSY, CPOSZ</position>
  <look_at>LCTX, LKY, LKTZ</look_at>
</camera>
<light>
  <position>LPOSX, LPOSY, LPOSZ</position>
  <color>LCOLR, LCOLG, LCOLB, LCOLA</color>
</light>
...
```

Ensuite, la caméra qui va capturer l'image de la scène est définie par sa position dans l'espace (CPOSX, CPOSY, CPOSZ) et le point de la scène qu'elle observe (LCTX, LKY, LKTZ). Cette caméra étant une caméra *trou d'épingles* on peut définir sa position par un

seul point de l'espace. Pour le mode *rt* (*ray tracing récursif*) les lumières sont définies comme des sources ponctuelles (A.2) par une seule position (LPOSX, LPOSY, LPOSZ) et une couleur RGB (LCOLR, LCOLG, LCOLB, LCOLA).

Les objets de la scène (sphères et plans) sont définis comme en A.3. Un plan par sa normale (NORMX, NORMY, NORMZ) et sa distance à l'origine D, une sphère par la position de son centre (CENTX, CENTY, CENTZ) et son rayon R. Les objets possèdent une couleur (COLR, COLG, COLB, COLA) ou une DSP REFL_SPD selon le mode de rendu utilisé.

Code A.3 – Définition d'un plan et d'une sphère

```
...
<plane>
  <normal>NORMX, NORMY, NORMZ</normal>
  <distance>D</distance>
  <color>COLR, COLG, COLB, COLA</color> <!-- ray/path tracing -->
  <reflectance>REFL_SPD</reflectance> <!-- spectral path tracing -->
  <material>MAT</material>
</plane>
<sphere>
  <center>CENTX, CENTY, CENTZ</normal>
  <radius>R</distance>
  <color>COLR, COLG, COLB, COLA</color> <!-- ray/path tracing -->
  <reflectance>REFL_SPD</reflectance> <!-- spectral path tracing -->
  <material>MAT</material>
</sphere>
```

Dans le cas des objets qui sont des sources de lumières il faut ajouter un paramètre EMS pour l'émission de la source, qui est soit une valeur flottante dans les cas du ray tracing et du path tracing soit une DSP représentant le spectre d'émission de la source pour le spectral path tracing. Exemple en A.4.

Code A.4 – Définition d'un objet-source

```
...
<sphere>
  ...
  <emission>EMS</emission>
</sphere>
```

Enfin, pour un objet fluorescent, il faut définir le spectre d'absorption de l'objet ABS_SPD et la valeur S du déplacement de stokes de celui-ci.

Code A.5 – Définition d'un objet fluorescent

```
...
<sphere>
  ...
  <fluorescence>
    <absorption>ABS_SPD</absorption>
    <stokes>S</stokes>
  </fluorescence>
</sphere>
```

Code A.6 – Exemple complet d'une scène présentant certains des effets implémentés (spectral path tracing/matériaux différents/illuminant D65/fluorescence/...)

```

<scene>
  <settings>
    <xres>400</xres>
    <yres>225</yres>
    <path>test_smcpt_d65_fluo_4096sp.bmp</path>
    <mode>spectral_mcpt</mode>
    <depth>10</depth>
    <sampl>4096</sampl>
    <illum>bbd65</illum>
  </settings>
  <camera>
    <position>0.000000, 5.000000, -6.000000</position>
    <look_at>0.000000, 2.000000, 2.000000</look_at>
  </camera>
  <plane>
    <normal>0.000000, 0.000000, -1.000000</normal>
    <distance>-10</distance>
    <material>matte</material>
    <reflectance>gray</reflectance>
    <emission>0.0</emission>
  </plane>
  <plane>
    <normal>0.000000, 1.000000, 0.000000</normal>
    <distance>0</distance>
    <material>matte</material>
    <reflectance>const</reflectance>
    <emission>0.0</emission>
  </plane>
  <sphere>
    <center>0.000000, 1.500000, 1.000000</center>
    <radius>1.5</radius>
    <material>diamond</material>
    <reflectance>const</reflectance>
    <emission>0.0</emission>
  </sphere>
  <sphere>
    <center>4.000000, 1.800000, 3.500000</center>
    <radius>1.8</radius>
    <material>metal</material>
    <reflectance>orange</reflectance>
    <emission>0.0</emission>
    <fluorescence>
      <absorption>blue</absorption>
      <stokes>220</stokes>
    </fluorescence>
  </sphere>
  <sphere>
    <center>-2.500000, 2.500000, 4.000000</center>
    <radius>2.5</radius>
    <material>metal</material>
    <reflectance>blue</reflectance>
    <emission>0.0</emission>
    <fluorescence>
      <absorption>purple</absorption>
      <stokes>120</stokes>
    </fluorescence>
  </sphere>

```

```
</fluorescence>
</sphere>
<sphere>
  <center>11.000000, 14.000000, -2.500000</center>
  <radius>6</radius>
  <material>matte</material>
  <emission>bbd65</emission>
</sphere>
</scene>
```

Annexe B

Ray tracing

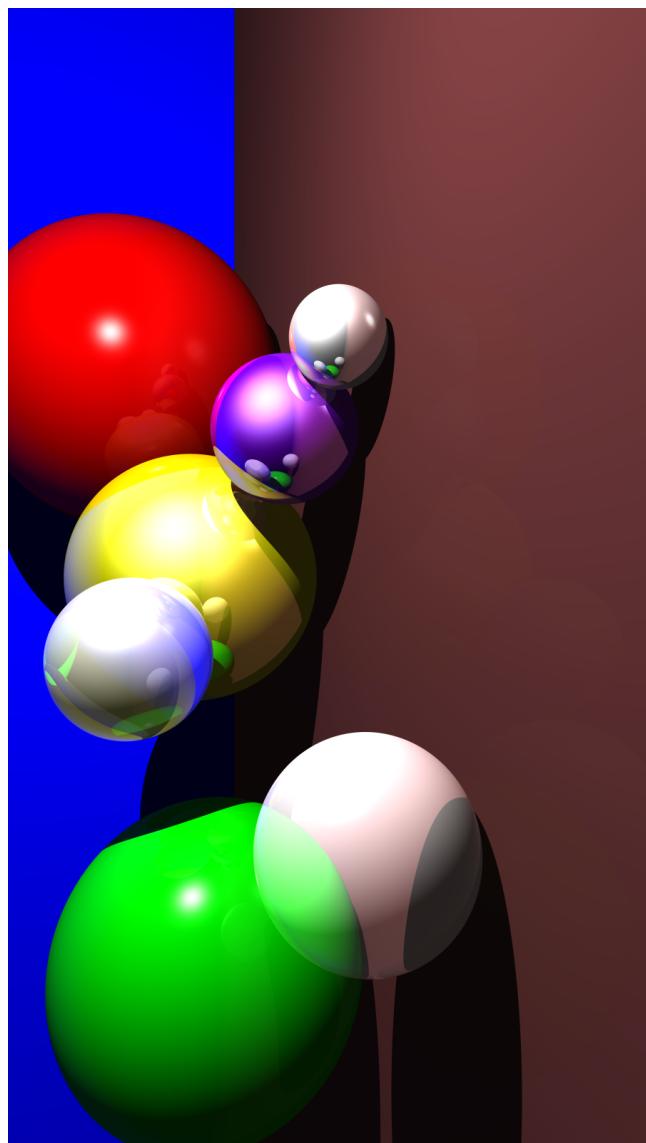


FIGURE B.1 – Exemple de scène rendue avec l'algorithme de ray tracing récursif, avec réflexions, réfractions avec différents indices, antialiasing,... À noter que les objets transparents projettent des ombres.

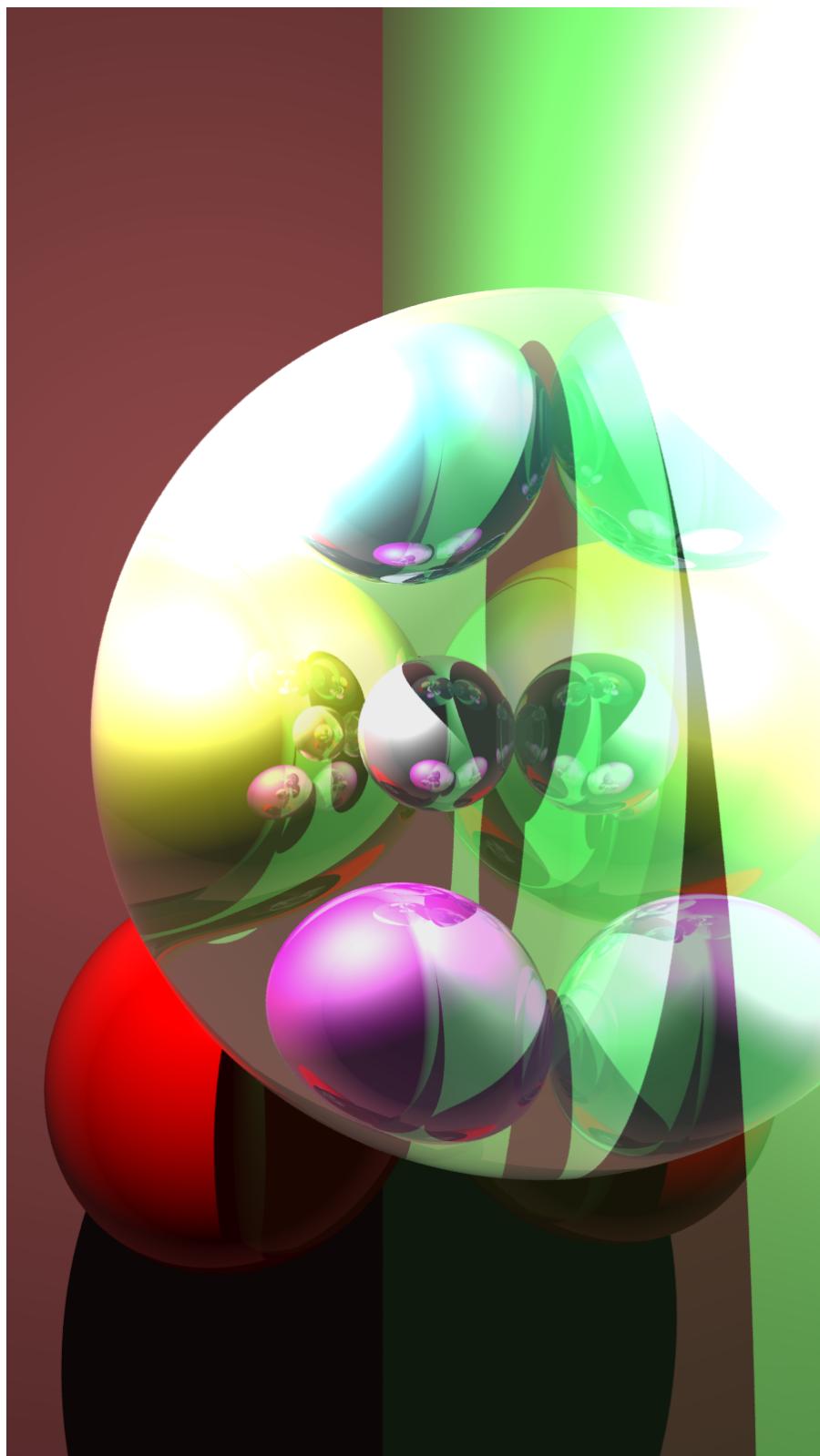


FIGURE B.2 – Exemple de scène rendue avec l'algorithme de ray tracing récursif. Quatre sphères (blanche, cyan, jaune et magenta) sont placées sous un dôme d'indice de réfraction 2.4. La sphère jaune semble bien plus volumineuse que les autres, ce n'est pas le cas en réalité. Aussi, la sphère rouge n'est pas visible derrière le dôme.

Annexe C

Path tracing

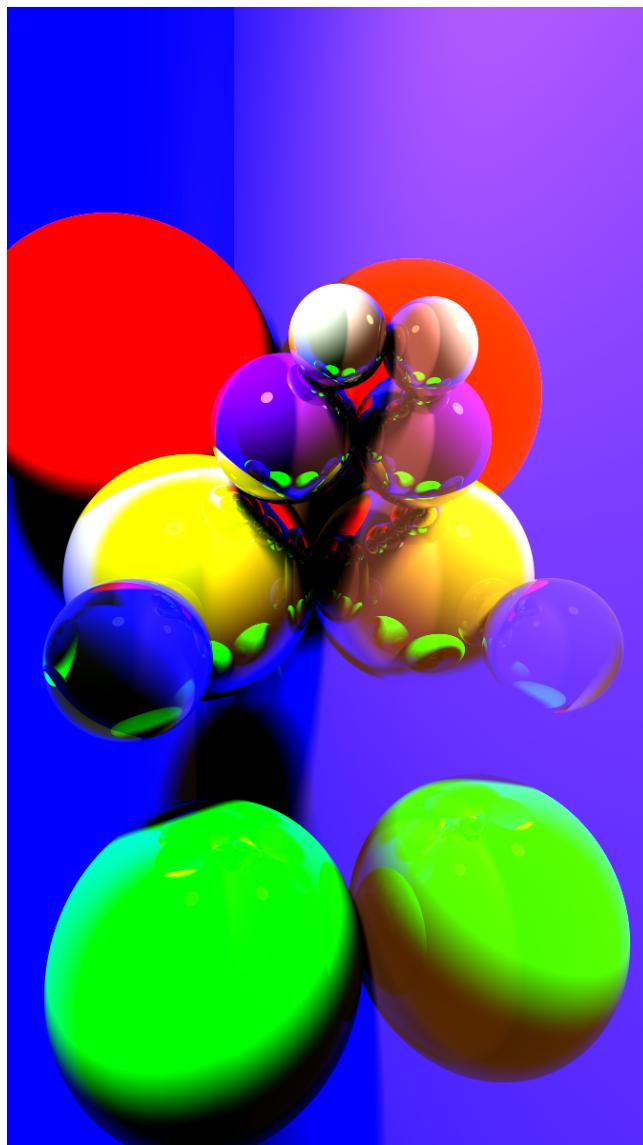


FIGURE C.1 – Exemple de scène rendue avec l'algorithme de path tracing, avec échantillonnage explicite des sources. À noter la présence de *color bleeding*, pénombre, etc.

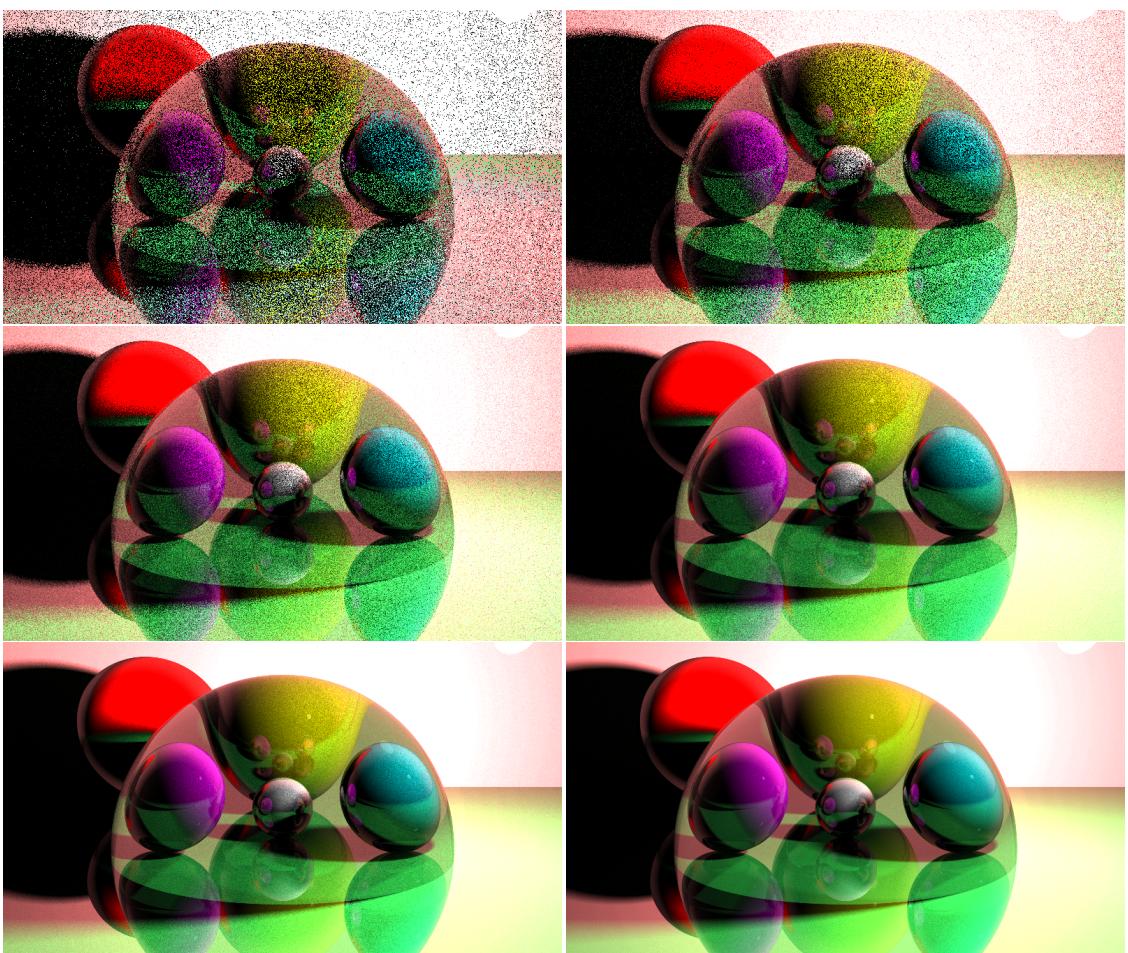


FIGURE C.2 – Scène rendue en mode path tracing avec des couleurs RGB et un différent nombre d'échantillons par pixel (1 4 16 64 256 1024). À noter les différences avec la version ray tracing de la même scène.

Annexe D

Spectral path tracing

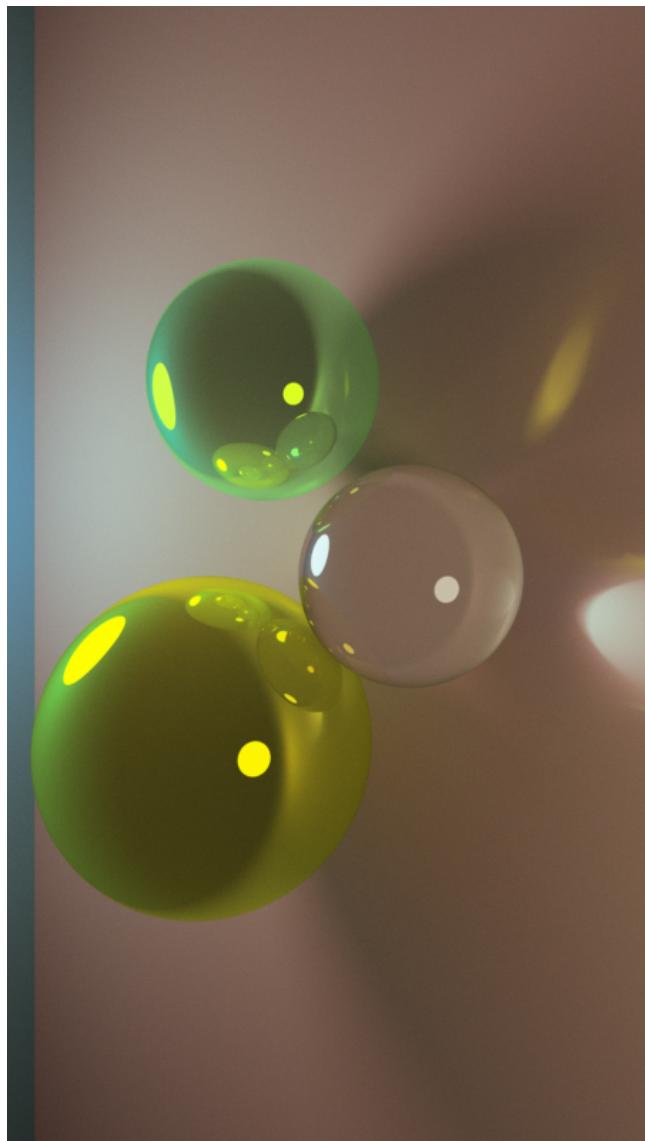


FIGURE D.1 – Exemple de scène rendue avec l'algorithme de spectral path tracing. Deux sources sont présentes dans la scène, une importante qui provoque ombres et caustique et une petite éloignée derrière la caméra.

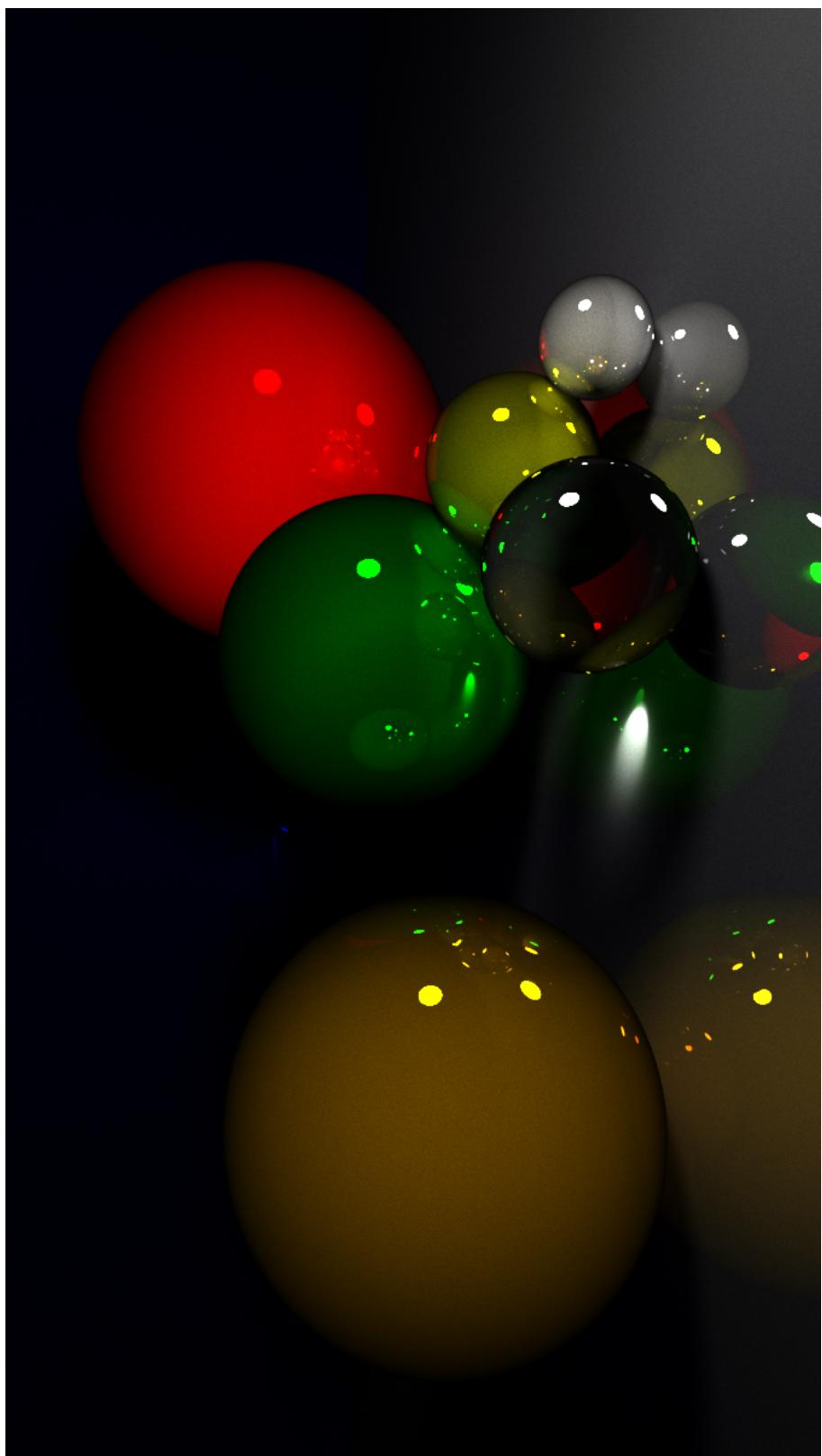


FIGURE D.2 – Exemple de scène rendue avec l'algorithme de spectral path tracing. La source est petite et éloignée, il a donc fallu un très grand nombre d'échantillons pour rendre cette scène : 262144

Annexe E

Fluorescence

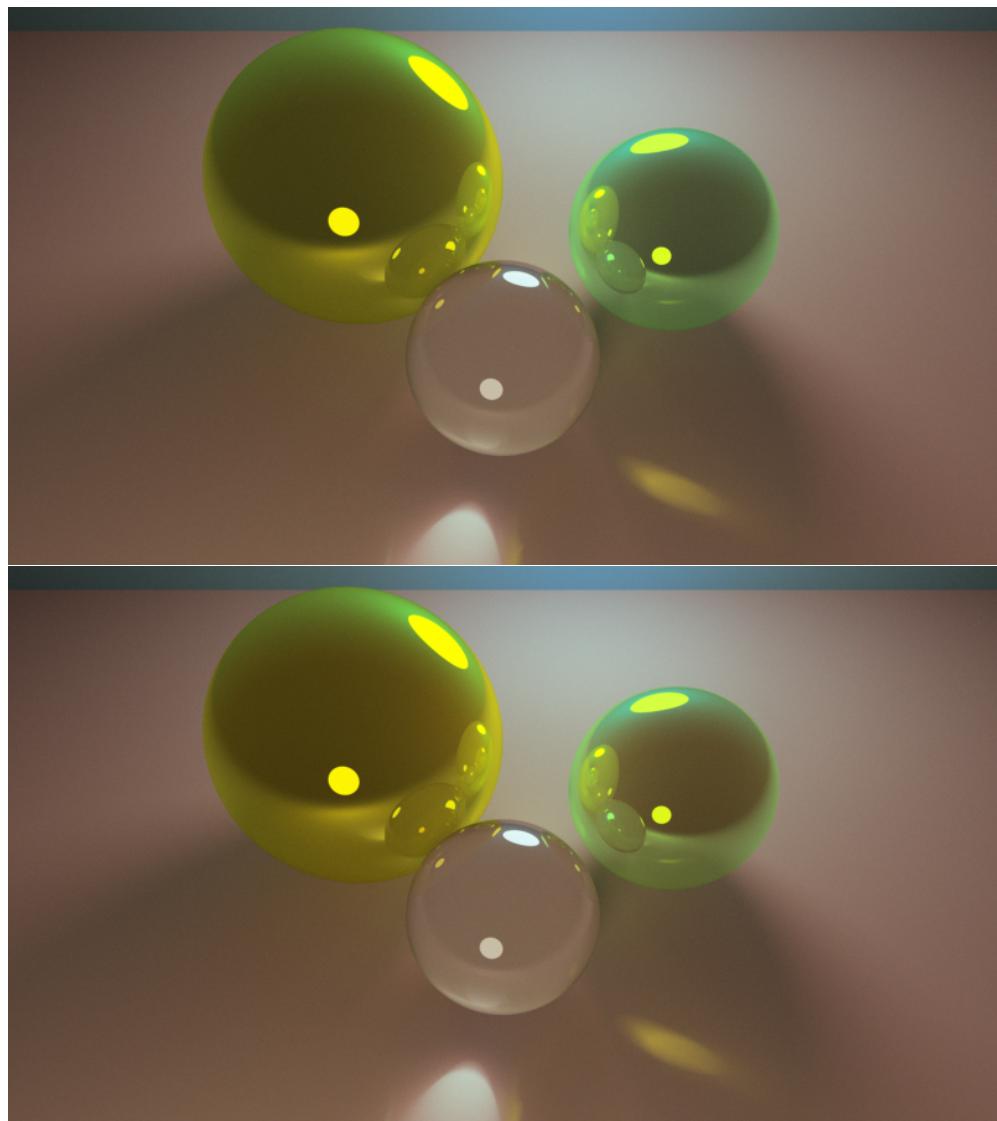


FIGURE E.1 – Comparaison entre la scène présentée en D.1 et sa version fluorescente (les sphères jaune et verte sont toutes les deux fluorescentes, absorbent dans les ultra-violets et émettent respectivement dans les rouges et dans les oranges). Nombre d'échantillons : 4096

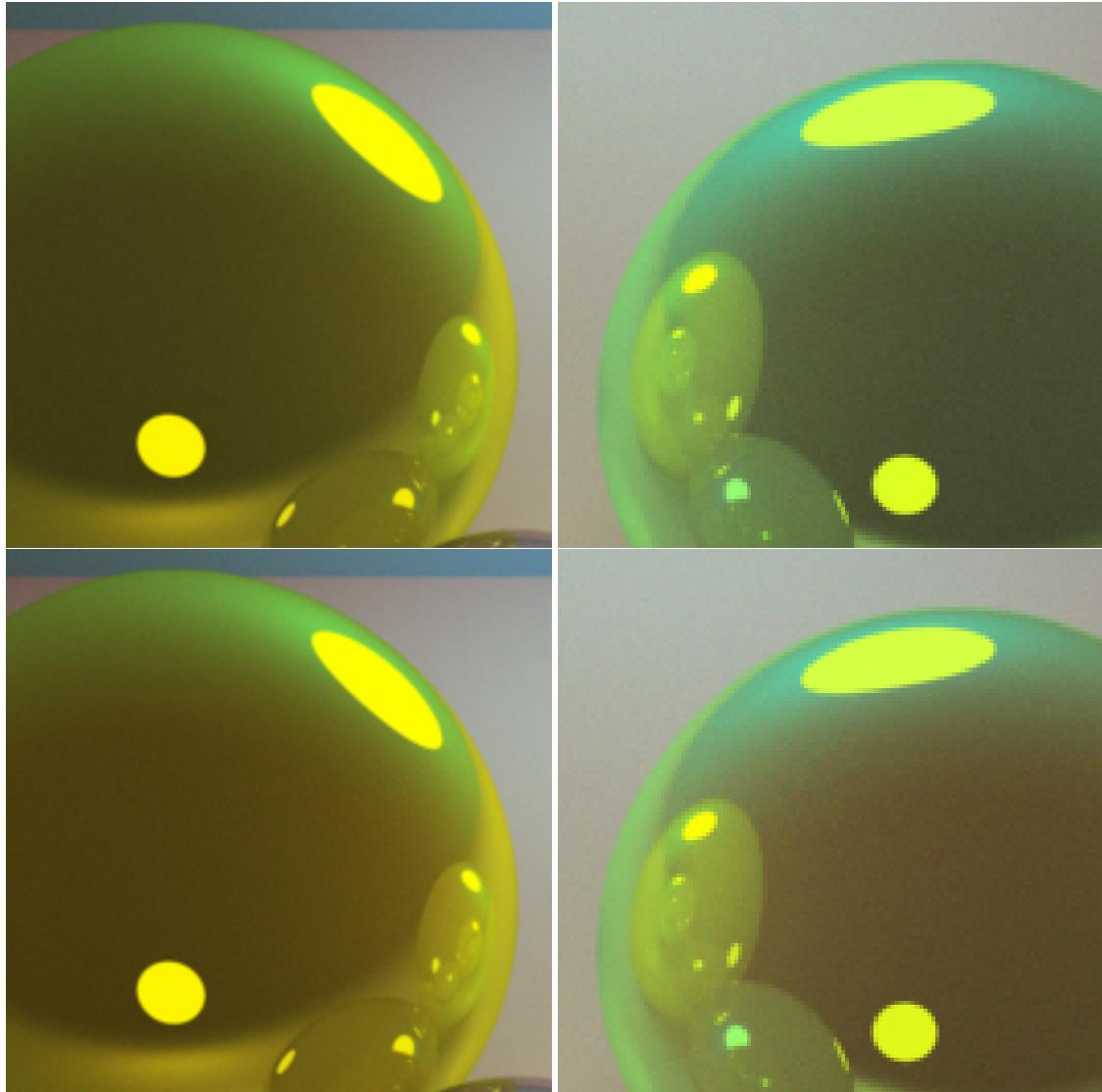


FIGURE E.2 – Détails de la scène pour une meilleure visibilité de l'effet de fluorescence (images de la ligne du bas)

Bibliographie

- [Abd+17] Marwan ABDELLAH et al. « A Physically Plausible Model for Rendering Highly Scattering Fluorescent Participating Media ». In : *arXiv preprint : arXiv :1706.03024v2* (juin 2017) (cf. p. 10, 35).
- [App68] Arthur APPEL. « Some Techniques for Shading Machine Renderings of Solids ». In : *Spring Joint Computer Conference* (1968), p. 37–45 (cf. p. 14).
- [Bli77] James F. BLINN. « Models of Light Reflection for Computer Synthesized Pictures ». In : *SIGGRAPH Comput. Graph.* 11.2 (juil. 1977), p. 192–198 (cf. p. 15).
- [HDF15] Johannes HANIKA, Marc DROSKE et Luca FASCIONE. « Manifold Next Event Estimation ». In : *Computer Graphics Forum* (2015) (cf. p. 28).
- [HH65] J. M. HAMMERSLEY et D. C. HANDSCOMB. *Monte Carlo methods*. Repr. with minor corrections. London : Chapman et Hall, 1965 (cf. p. 16).
- [ICG86] David S. IMMEL, Michael F. COHEN et Donald P. GREENBERG. « A Radiosity Method for Non-diffuse Environments ». In : *SIGGRAPH Comput. Graph.* 20.4 (août 1986), p. 133–142 (cf. p. 11).
- [JA18] Adrian JARABO et Victor ARELLANO. « Bidirectional Rendering of Vector Light Transport ». In : *Computer Graphics Forum* 37.6 (2018), p. 96–105 (cf. p. 40).
- [Kaj86] James T. KAJIYA. « The Rendering Equation ». In : *SIGGRAPH Comput. Graph.* 20.4 (août 1986), p. 143–150 (cf. p. 11, 15, 26).
- [McN+06] George McNAMARA et al. « Spectral imaging microscopy web sites and data ». In : *Cytometry Part A* 69A.8 (2006), p. 863–871 (cf. p. 32).
- [MFW18] Michal MOJZÍK, Alban FICHET et Alexander WILKIE. « Handling Fluorescence in a Uni-directional Spectral Path Tracer ». In : *Computer Graphics Forum* 37.4 (2018), p. 77–94 (cf. p. 40).
- [MU12] Rosana MONTES SOLDADO et Carlos UREÑA ALMAGRO. « An Overview of BRDF Models ». In : <http://digibug.ugr.es/handle/10481/19751> (mar. 2012) (cf. p. 18).
- [Nar+06] Srinivasa G. NARASIMHAN et al. « Acquiring Scattering Properties of Participating Media by Dilution ». In : *ACM Trans. Graph.* 25.3 (juil. 2006), p. 1003–1012 (cf. p. 35).
- [Nic65] Fred E. NICODEMUS. « Directional Reflectance and Emissivity of an Opaque Surface ». In : *Appl. Opt.* 4.7 (juil. 1965), p. 767–775 (cf. p. 12).
- [Pau17] Mathias PAULIN. « Modélisation et simulation de l'éclairage ». In : *Aspects théoriques de l'informatique graphique - Partie Rendu* (2017) (cf. p. 16).
- [PH10] Matt PHARR et Greg HUMPHREYS. *Physically Based Rendering : From Theory to Implementation*. 2nd ed. Morgan Kaufmann, 2010 (cf. p. 9, 19).
- [Rei+02] Erik REINHARD et al. « Photographic Tone Reproduction for Digital Images ». In : *ACM Trans. Graph.* 21.3 (juil. 2002), p. 267–276 (cf. p. 20).
- [Rus88] Holly RUSHMEIER. « Realistic image synthesis for scenes with radiatively participating media ». Thèse de doct. Cornell University, 1988 (cf. p. 10).

- [S94] Andrew S. GLASSNER. « A Model for Fluorescence and Phosphorescence ». In : *Proceedings Fifth Eurographics Workshop on Rendering*. Springer Berlin Heidelberg, juin 1994, p. 57–68 (cf. p. 21).
- [Sch94] Christophe SCHLICK. « An Inexpensive BRDF Model for Physically-based Rendering ». In : *Computer Graphics Forum* 13.3 (août 1994), p. 233–246 (cf. p. 24).
- [Sto52] George Gabriel STOKES. « XXX. On the change of refrangibility of light ». In : *Philosophical Transactions of the Royal Society of London* 142 (1852), p. 463–562 (cf. p. 20).
- [TJ31] Smith T. et Guild J. « The C.I.E. colorimetric standards and their use ». In : *Transactions of the Optical Society* 33.3 (1931), p. 73 (cf. p. 19).
- [Whi80] Turner WHITTED. « An Improved Illumination Model for Shaded Display ». In : *Commun. ACM* 23.6 (juin 1980), p. 343–349 (cf. p. 15, 24).
- [Wil+06] Alexander WILKIE et al. « A Reflectance Model for Diffuse Fluorescent Surfaces ». In : *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques*. GRAPHITE '06. ACM, 2006, p. 321–331 (cf. p. 22).
- [ZSS14] Yinqiang ZHENG, Imari SATO et Yoichi SATO. « Spectra Estimation of Fluorescent and Reflective Scenes by Using Ordinary Illuminants ». In : *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, p. 188–202 (cf. p. 31).