

# Structures pour l'édition de phénomènes d'éclairage

Romain Pacanowski

sous la direction conjointe de

Pierre Poulin, Professeur à l'Université de Montréal

Xavier Granier, Chargé de Recherche à l'INRIA

Christophe Schlick, Professeur à l'Université de Bordeaux

Département d'informatique et de recherche opérationnelle  
Université de Montréal

*Présenté dans le cadre de  
l'examen oral prédoctoral*

Jury

Victor Ostromoukhov	(président)
Pierre Poulin	(directeur)
Yann-Gaël Guéhéneuc	(membre)

**13 septembre 2007**

© Romain Pacanowski

# Résumé

Ces vingt dernières années, la synthèse d'images a effectué un bond spectaculaire dans la génération d'images photo-réalistes. Cette qualité est rendue possible en partie grâce au développement d'algorithmes de simulation de l'éclairage et de modèles de réflectance toujours plus précis dans la représentation des matériaux et leur interaction avec la lumière. Cependant, le contrôle de l'apparence de l'image synthétisée reste une tâche toujours autant difficile à accomplir, compte tenu des paramètres plus nombreux, complexes, interconnectés, et d'effets hautement non-linéaires. Le besoin d'édition artistique apparaît pourtant dans de nombreux domaines dont celui des films d'animation, des jeux vidéo et de la publicité. Les retouches/corrections souhaitées par un artiste peuvent avoir une réalisation physique ou encore être complètement irréalistes. Dans cette recherche, nous proposons de développer des systèmes interactifs d'édition (directe et inverse) de l'apparence en nous concentrant sur deux phénomènes essentiels et omniprésents d'éclairage : la réflectance des matériaux (BRDF) et l'éclairage indirect.

Notre concept central est de relâcher lorsque nécessaire certaines contraintes physiques afin d'autoriser plus de flexibilité et de créativité à l'artiste, mais tout en forçant l'édition à garder une cohérence lors du déplacement d'objets, de sources de lumière ou du point vue. Nous introduisons plusieurs paradigmes d'édition, abstractions, interfaces adaptées et structures d'accélération pour rendre intuitive, flexible et efficace l'édition interactive de ces deux phénomènes.

## Mots clefs :

Apparence de surface, BRDF, éclairage global, rendu inverse, *caching*.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Synthèse d'images	1
1.2	Éclairage	1
1.3	Édition	2
1.4	Proposition	3
1.5	Organisation du document	4
<b>2</b>	<b>Édition de BRDF</b>	<b>5</b>
2.1	La BRDF	5
2.1.1	Généralités	5
2.1.2	Définition mathématique	6
2.1.3	Aperçu de la recherche sur les BRDFs	10
2.1.4	Travaux antérieurs sur l'édition interactive de BRDF	14
2.2	Recherche proposée en édition	17
2.2.1	Motivation	17
2.2.2	Paramétrisations de la BRDF	17
2.2.3	Paradigme d'édition orienté dessin vectoriel	20
2.2.4	Paradigme orienté image	24
<b>3</b>	<b>Structure pour l'éclairage indirect</b>	<b>27</b>
3.1	Introduction à l'éclairage global en synthèse d'images	27
3.1.1	Approche intuitive	27
3.1.2	Formulation mathématique du rendu	27
3.1.3	Éclairage indirect	28
3.1.4	Aperçu des algorithmes d'éclairage global	29
3.1.5	Notion de <i>caching</i>	30
3.2	Problématique et travaux antérieurs en <i>caching</i>	31
3.3	Recherche proposée en <i>caching</i>	34
3.3.1	Aperçu général de la solution proposée	34
3.3.2	Construction de la structure	36
3.3.3	Reconstruction de l'éclairage indirect	38
3.3.4	Analyse de l'erreur	39
3.3.5	Comparaisons avec les techniques antérieures	39
3.4	Travaux antérieurs en édition d'éclairage	40
3.5	Recherche proposée en édition	42

3.5.1	Similarités des deux paradigmes . . . . .	42
3.5.2	Paradigme d'édition directe . . . . .	43
3.5.3	Paradigme d'édition orientée peinture . . . . .	44
3.5.4	Interaction après édition . . . . .	44
<b>4</b>	<b>Échéancier</b>	<b>47</b>
<b>5</b>	<b>Conclusion et travaux futurs</b>	<b>49</b>
5.1	Conclusion . . . . .	49
5.2	Travaux futurs . . . . .	49
	<b>Bibliographie</b>	<b>51</b>
<b>A</b>	<b>Radiométrie et Photométrie</b>	<b>57</b>
A.1	Quantités radiométriques . . . . .	57
A.2	Définition quantique de l'énergie rayonnante spectrale . . . . .	62
A.3	Définition du flux énergétique spectral en fonction de la température . . . . .	62
<b>B</b>	<b>Une base alternative pour reconstruire l'irradiance dans l'espace des directions</b>	<b>65</b>
B.1	Grandeurs radiométriques vectorielles . . . . .	65
B.2	Base de fonctions sur la sphère . . . . .	66
B.3	Une base pour les vecteurs d'irradiance . . . . .	67
B.4	Retour sur la proposition de la section 3.3.3 . . . . .	69
<b>C</b>	<b>Article de travail sur l'édition vectorielle de BRDF</b>	<b>71</b>
<b>D</b>	<b>Article de travail sur la structure de cache</b>	<b>79</b>

# Table des figures

2.1	Exemple de matériaux représentables par une SVBRDF. Images tirées de [LBAD <sup>+</sup> 06].	6
2.2	Un exemple de matériau représentable par une BTF. Images tirées de [MMS <sup>+</sup> 06]. . .	6
2.3	Repérage des directions pour la BRDF. La direction incidente $\omega_i$ de la lumière et la direction d'observation $\omega_o$ sont repérées dans le repère local $(x, B, T, N)$ de la surface. $\omega_i$ (resp. $\omega_o$ ) subtend un angle solide différentiel $d\omega_i$ (resp. $d\omega_o$ ) sur l'hémisphère $\Omega_N$ . . . . .	7
2.4	Illustration de la propriété d'isotropie et d'anisotropie d'une BRDF. (a) BRDF isotropique, BRDF anisotropique dont la direction principale d'anisotropie est (b) verticale ou (c) horizontale. Images tirées de [AS00]. . . . .	8
2.5	Construction d'une BRDF complexe à partir de BRDFs simples. De gauche à droite : La BRDF finale, la BRDF lambertienne, la BRDF spéculaire parfaite et la BRDF <i>glossy</i> . . . . .	9
2.6	Illustration de l'effet de Fresnel sur un matériau comme le bois. (a) La table en bois vue à incidence normale, le coefficient diffus domine le coefficient spéculaire. (b) Pour une incidence à 45 degrés, les coefficients sont plus ou moins égaux. (c) A incidence rasante le coefficient spéculaire domine. Photographies tirées de [LFTG97]. . . . .	11
2.7	Comparaison entre le modèle de Ward [War92] et le modèle d'Ashikhmin et al. [APS00]. Images tirées de [APS00]. . . . .	11
2.8	L'effet d'ombrage des microfacettes. En fonction de la direction de vue ( $\omega_o$ ) et de la direction de la lumière ( $\omega_i$ ), une proportion des microfacettes ne sont pas actives : elles sont masquées par d'autres microfacettes. . . . .	12
2.9	Exemple de gonioréfectomètre. Image tirée de [LFTW05]. . . . .	12
2.10	<b>bv</b> le logiciel de visualisation de BRDFs développé par Rusinkiewicz. . . . .	13
2.11	Illustration du système d'édition, à l'aide de courbes, de BRDFs de Ben-Artzi et al. Image tirée [BAOR06]. . . . .	15
2.12	Utilisation d'une texture pour définir la distribution des microfacettes du modèle d'Ashikhmin et al.[APS00]. . . . .	15
2.13	<b>BRDF Shop</b> . Images de gauche : le résultat de l'édition de la BRDF sur un objet. Images de droite : la sphère que l'utilisateur édite pour créer sa BRDF représentée en bas à droite. En haut, une édition de BRDF avec un seul lobe. En bas, édition d'une BRDF avec trois lobes. Images tirées de [CPK06]. . . . .	16
2.14	Les différents types de paramétrisation. (a) Classique, (b) Rusinkiewicz [Rus98], (c) la nôtre. . . . .	18

2.15	Paradigme d'édition vectorielle. Principe de l'édition du lobe. Les points de contrôle sont affichés en vert. (a) Définition par l'utilisateur de la courbe $\mathcal{C}(\phi_s)$ qui spécifie la forme du lobe. (b) Définition de la fonction de potentiel $\mathcal{P}$ , ici choisie de manière linéaire. (c) Choix de la couleur qui peut être fonction de $\rho$ et $\phi_s$ . (d) Lobe final obtenu.	20
2.16	Exemple d'un triangle sphérique défini par 3 configurations : au sommet de la normale où $(\theta_i, \phi_i) = (0, 0)$ , à angle rasant <i>i.e.</i> $(\theta_i, \phi_i) = (\frac{\pi}{2}, 0)$ , et enfin dans la direction principale d'anisotropie où $(\theta_i, \phi_i) = (\frac{\pi}{2}, \frac{\pi}{2})$ .	23
2.17	Exemple de gradient de couleur obtenu en spécifiant une couleur du lobe jaune pour la configuration $(\theta_i, \phi_i) = (0, 0)$ et une couleur rouge pour la configuration $(\theta_i, \phi_i) = (\frac{\pi}{2}, 0)$ .	24
2.18	<i>Highlights</i> dessinés avec un courbe de type (a) <i>star-shape</i> , (b) une ellipse, (c) la fonction analytique 4-pétales et une rétro-réflexion, (d) la fonction analytique 10-pétales.	24
3.1	Différence entre éclairage local et éclairage global. (a) Eclairage local. (b) Eclairage global. Images tirées de [htt05].	28
3.2	Rendus avec <i>photon mapping</i> . (a) Utilisation directe du <i>photon map</i> . (b) Utilisation du <i>final gathering</i> . Images tirées de [Jen96].	32
3.3	Comparaison entre de l'éclairage indirect reconstruit avec : (a) gradients de Ward (b) Approximation de meilleure qualité obtenue en échantillonnant l'hémisphère. Images tirées de [KGPB05].	33
3.4	Comparaison du positionnement des échantillons dans le cache en fonction du critère utilisé. (a) Critère de Tabellion et Lamorlette en utilisant la géométrie détaillée. (b) Critère de Tabellion et Lamorlette en utilisant la géométrie faiblement détaillée. (c) Critère de Ward. Images tirées de [TL04].	33
3.5	Comparaison de la qualité visuelle obtenue entre : (a) <i>caching</i> de Ward et al. [WH92]. (b) <i>caching</i> de Krinane [KBPv06]. Images tirées de [KBPv06].	34
3.6	Illustration du type de structures utilisées en fonction du type d'objet d'une scène. En plus de la structure globale (en rouge) de la scène, des structures locales 3D (en vert) sont utilisés pour les objets volumineux tandis que des structures locales 2D sont utilisées pour les objets de types planaires.	35
3.7	Illustration 2D du préconditionnement des valeurs à la frontière de la structure. La boîte englobante de la scène est représentée en traits épais noir. Les valeurs à l'intérieur de la boîte sont échantillonnées à l'aide d'un algorithme d'éclairage global. Les valeurs à l'extérieur sont par défaut nulles mais une valeur initiale leur est attribuée (préconditionnement) pour garantir une interpolation continue.	37
3.8	Préconditionnement des valeurs internes à un objet opaque.	39
3.9	Edition de l'éclairage indirect dans le cas d'une BRDF <i>glossy</i> . Pour faciliter la tâche d'édition de l'utilisateur, le cône de directions du lobe spéculaire de la BRDF sous-jacente est affiché. L'utilisateur peut donc facilement modifier le vecteur d'irradiance $I_N(x)$ et l'amener dans ce cône pour que son édition ait un impact sur l'apparence finale du point $x$ .	43
A.1	(a) Notion de flux. (b) Repérage de la radiance. La radiance est un scalaire valable pour un angle solide $d\omega$ repéré par une direction $\omega$ .	57
A.2	Reperage hémisphérique des directions.	58
A.3	Angle solide substanté par une surface $A$ au point $x$ .	59

A.4	(a) Angle solide différentiel $d\omega$ pour une direction $\omega$ . (b) Quantités en jeu pour la formulation surfacique de l'angle solide différentiel $d\omega_x$ . . . . .	60
A.5	Quantités radiométriques et photométriques . . . . .	61





# Chapitre 1

## Introduction

### 1.1 Synthèse d'images

En infographie, la réalisation d'une image<sup>1</sup> de synthèse nécessite plusieurs données regroupées sous le nom de scène. Une scène est en général composée de :

- une caméra
- une définition de la géométrie
- une définition des matériaux
- une définition des sources de lumière.

La caméra est un modèle mathématique qui permet de visualiser en 2D les éléments contenus dans la scène 3D. Elle est au moins composée d'un point de vue, d'une direction de vue et de la résolution de l'image à générer. Cette caméra minimale peut être assimilée à un appareil photographique où tout le contenu de la scène est au focus. L'ensemble des objets contenus dans la scène forment sa géométrie, définie à l'échelle macroscopique. Il existe plusieurs manières de spécifier la géométrie tels les maillages polygonaux, les surfaces paramétriques, les surfaces implicites, les surfaces de subdivision ou encore les ensembles de points. Les matériaux de la scène sont appliqués sur la géométrie et spécifient la façon dont les objets affectent la lumière. Enfin, la dernière donnée nécessaire est la spécification des sources de lumière, car il ne peut y avoir d'image sans source de lumière. Une source de lumière est définie par une géométrie (la forme) et par une fonction d'émission. Cette fonction décrit la quantité d'énergie (puissance), la fréquence et la distribution spatiale des particules lumineuses émises. Un algorithme de rendu prend ainsi en entrée une scène et génère une image de synthèse.

### 1.2 Éclairage

On distingue, d'ordinaire, les algorithmes de rendu utilisant un modèle d'éclairage local de ceux utilisant un modèle d'éclairage global. En éclairage local, les objets ne sont éclairés que directement par les sources de lumière, tandis qu'en éclairage global tout objet peut potentiellement réfléchir la lumière vers d'autres objets. L'éclairage global permet donc de prendre en compte l'ensemble des chemins lumineux. Un objet est donc éclairé par les sources de lumière (éclairage direct) et par les autres objets de la scène (éclairage indirect). L'éclairage local peut être considéré comme une composante de l'éclairage global. Quelque soit le type d'algorithme de rendu utilisé, l'apparence

---

<sup>1</sup>On entend par image un tableau bidimensionnel dont chaque élément, appelé pixel, encode une couleur.

finale de l'image produite dépend de l'ensemble des paramètres de la scène. Lorsque l'on utilise un modèle d'éclairage local, il existe une dépendance directe entre les paramètres et le résultat obtenu. Souvent, l'éclairage direct constitue la contribution la plus significative dans l'apparence finale d'une image. En éclairage global, la dépendance des paramètres sur le résultat est en général nettement plus compliquée parce qu'elle est non-linéaire. Il est alors plus difficile de prévoir l'apparence finale de l'image lorsqu'on modifie l'un des paramètres de la scène.

## 1.3 Édition

Le besoin de contrôler ou d'éditer l'apparence d'une image est nécessaire dans la plupart des applications où l'esthétique de l'image revêt une importance certaine incluant, pour leur omniprésence, les domaines du jeu vidéo et des films d'animation. L'artiste,<sup>2</sup> qui n'est ni physicien, ni mathématicien, ni informaticien, souhaite souvent modifier l'apparence finale de l'image sur certains aspects que l'algorithme de rendu n'aura pas pris en compte en fonction de ses attentes.

Les modifications que l'artiste effectue sur l'apparence sont diverses et variées. Il peut modifier l'image de manière à ce qu'elle reste physiquement correcte ou physiquement plausible, mais il peut aussi choisir d'introduire des modifications qui sont physiquement impossibles. Le terme "physiquement correct" indique qu'il existe bien une configuration physique de la scène qui produira la nouvelle apparence. Un autre élément important de l'apparence est que les modifications effectuées par l'artiste devraient conserver un comportement "cohérent" lorsque l'on modifie d'autres paramètres tels que le point de vue ou encore les objets (déplacements ou changements de forme). Il n'est pas possible de donner une définition générale et précise du terme cohérent, mais nous clarifierons mieux ce que l'on entend par "cohérent" au fil du document et selon le contexte.

Illustrons à l'aide d'un exemple simple ces différentes notions. Lorsqu'un artiste insère une ombre dans une scène alors qu'aucun objet ne projette cette ombre, la modification qu'il porte est physiquement impossible.<sup>3</sup> Il en va de même la création d'une ombre de forme circulaire quand l'objet qui la génère est un cube. Une modification physiquement possible serait de modifier le placement d'une ombre en ajustant la position d'une lumière. Entre ces deux types de modification, une modification physiquement plausible serait d'étendre artificiellement la taille de la pénombre sans pour autant changer la taille de la surface de la source de lumière. On parle dans ce cas de modification physiquement plausible lorsqu'un observateur ne détectera pas la modification ; elle ne le choquera pas le moins du monde. La cohérence dans le contexte des ombres peut s'appliquer lorsque l'artiste modifie d'autres paramètres tels que le point de vue ou encore les objets (déplacements ou changements de forme). Par exemple, dans une séquence animée, l'artiste spécifie une forme d'ombre différente pour deux instants distincts. Il s'attend alors à ce que l'ombre varie de manière lisse entre ces deux instants. Dès lors, ce type de contrainte doit être prise en compte dans la représentation du phénomène que manipule l'artiste.

Pour éditer ou contrôler l'image produite, l'artiste peut soit manipuler les différents paramètres de la scène (édition directe), soit éditer directement l'apparence dans l'image (édition inverse). En édition directe, le nombre de paramètres tend souvent à être élevé, et par conséquent l'artiste peut mettre un certain temps à maîtriser leurs interdépendances. Par exemple, le modèle de réflectance de He et al. [HTSG91] contient de nombreux paramètres dont l'autocorrelation de la hauteur des

---

<sup>2</sup>Dans ce document, artiste, usager et utilisateur désignent la même entité.

<sup>3</sup>En cinématographie, ce type d'effet est parfois réalisé en plaçant hors du champ de la caméra un objet et une source de lumière très concentrée.

bosses microscopiques de la surface, un terme de masquage ou encore un coefficient de Fresnel qui sont très difficiles à maîtriser pour un utilisateur novice parce qu'ils ont une action non-linéaire sur l'apparence. L'édition inverse est plus immédiate pour l'artiste qui "peint" simplement la couleur dans les pixels, mais elle nécessite de retrouver les paramètres qui produiront le résultat escompté. Il est aussi nécessaire de comprendre les intentions de l'artiste, parce que bien souvent il n'existe pas de solution unique, voire aucune, à ses attentes. De plus, l'artiste peut spécifier sans s'en rendre compte ou intentionnellement des attentes contradictoires du point de vue du système !

Un besoin aussi bien en édition inverse qu'en édition directe, est de mettre à jour le résultat des modifications en temps interactif, et dans l'idéal en temps réel. Cette interactivité est nécessaire pour donner l'impression à l'artiste qu'il effectue un travail d'art, comme il le ferait avec un crayon sur une feuille de dessin ou avec un burin sur une pierre. Quelque soit la forme que prend l'édition, le nombre de paramètres, leur haute dimensionnalité et leur interdépendance nécessitent soit de réduire leur dimension, soit leur nombre afin de pouvoir les éditer.

## 1.4 Proposition

Dans cette proposition de doctorat nous souhaitons nous concentrer sur l'édition de deux facettes importantes de l'apparence :

- l'édition au niveau microscopique du matériau représenté par la fonction de réflectance (BRDF)
- l'édition de l'éclairage indirect.

Il nous semble important de travailler sur l'édition de la BRDF étant donnée sa place centrale non seulement dans les algorithmes de rendu, mais aussi dans des domaines connexes comme l'échantillonnage. La BRDF constitue l'un des sujets de recherche en infographie les plus étudiés ces vingt dernières années pour la définir et la mesurer. Le développement d'algorithmes prenant en compte l'éclairage indirect est aussi un axe important de la recherche en infographie parce qu'ils permettent d'obtenir des images contenant des effets plus réalistes : les objets apparaissent comme pleinement immergés dans leur environnement. Cette quête du réalisme se poursuit toujours, bien que maintenant la majorité des efforts portent sur l'amélioration de l'efficacité des algorithmes.

Notre postulat central est de bien encadrer les attentes potentielles d'un artiste et de développer des structures appropriées qui lui permettront d'éditer artistiquement une apparence. Nous sommes prêts à relâcher certaines contraintes physiques ou algorithmiques liées aux phénomènes modélisés afin de permettre une plus grande expressivité artistique. Cependant, nous souhaitons conserver un comportement cohérent lors des actions d'édition et du rendu du résultat. De plus, nous fixons à notre développement les contraintes suivantes :

1. Le système doit répondre de manière interactive aux commandes d'édition.
2. L'interface utilisateur doit rester simple et intuitive.
3. Le type d'édition peut aller du physiquement correct au physiquement impossible.
4. Le processus d'édition doit s'insérer dans un schéma cohérent par rapport aux configurations de lumière, de point de vue et des objets.

Bien que le concept de cohérence soit difficile à définir dans notre contexte nous nous servirons d'exemples du réel (photographies ou simulations par exemple) pour exhiber le comportement général du phénomène édité. Enfin, dans cette proposition nous développerons, pour chacun des phénomènes étudiés, un paradigme d'édition directe et un paradigme d'édition inverse.

## 1.5 Organisation du document

Dans le chapitre 2, nous nous intéressons à l'édition de la BRDF. Nous rappelons la définition mathématique de la BRDF ainsi que ses différentes formes de paramétrisation. Nous donnons un bref aperçu de la recherche sur le développement de modèles pour les BRDFs ainsi que sur les systèmes d'édition directe et inverse. Nous proposons deux paradigmes d'édition pour les BRDFs.

Le chapitre 3 introduit les notions fondamentales qui ont trait au calcul et à l'édition de l'éclairage global. Nous passons brièvement en revue les algorithmes d'éclairage global et mettons en avant l'importance du *caching*, principale technique d'accélération implémentée dans les logiciels. Dans ce contexte, nous proposons une nouvelle structure capable de mettre en cache l'éclairage indirect et de l'édition.

Un échéancier au chapitre 4 détaille les prochaines étapes de nos travaux et les forums auxquels nous soumettrons nos résultats. Nous finissons par résumer nos contributions et donner plusieurs pistes de recherche pour des travaux futurs.

La première annexe du document regroupe l'ensemble des notations physiques importantes en synthèse d'images et le lecteur est invité à s'y référer s'il souhaite obtenir plus de précisions sur la définition des quantités manipulées dans les deux chapitres principaux. Les deux annexes suivantes fournissent deux articles en version préliminaire auxquels nous ajouterons les prochains résultats.

# Chapitre 2

## Édition de BRDF

Dans ce chapitre nous introduisons le concept de BRDF et donnons un rappel des définitions mathématiques qui y sont associées. Parmi elles, nous détaillons la notion de paramétrisation qui est centrale aux BRDFs. Nous présentons les techniques d'édition de BRDF en mettant en avant leurs limitations avant de proposer deux paradigmes d'édition.

### 2.1 La BRDF

#### 2.1.1 Généralités

Lorsque l'on observe, dans les mêmes conditions d'éclairage, un mur peint à l'aide d'une peinture mate, nous ne le percevons pas de la même façon qu'un mur peint à l'aide d'une peinture satinée. Ce dernier nous paraît plus "brillant" tandis que le premier nous paraît plus sombre. La fonction de réflectance, appelée BRDF, modélise cette différence de comportement de la matière par rapport à la lumière.

D'un point de vue physique, la BRDF (*Bidirectional Reflectance Distribution Function*) modélise les interactions entre la lumière et la matière à l'échelle microscopique. En première approximation, pour une direction incidente de lumière, la BRDF représente la quantité d'énergie lumineuse<sup>1</sup> reçue dans une direction d'observation. Si l'on considère la lumière comme un signal, on peut interpréter que la BRDF agit comme un filtre angulaire [RH01] sur la lumière incidente (signal initial) en la transformant en lumière réfléchi (signal transformé).

En infographie, la matière est assimilable à un ensemble d'objets géométriques (échelle macroscopique). La BRDF modélise le comportement, vis-à-vis de la lumière, de l'objet en chacun de ses points infinitésimaux. La BRDF est donc reliée à la surface de l'objet, mais de manière locale, et elle est souvent supposée uniforme sur toute cette surface. Lorsque cette supposition est vérifiée, le matériau modélisé par la BRDF ne possède pas de détails individuellement observables.

Lorsque le matériau d'un objet n'est pas uniforme sur sa surface, on ne parle plus de BRDF mais de SVBRDF (*Spatially Varying BRDF*). Les images de la figure 2.1 montrent un exemple de matériau modélisé par une SVBRDF. Bien que la variation spatiale du matériau soit prise en compte, la SVBRDF reste une fonction qui se place à l'échelle microscopique et ne permet donc pas de prendre en compte des phénomènes qui surviennent à l'échelle mésoscopique<sup>2</sup>. Les effets

---

<sup>1</sup>Le lecteur soucieux d'une définition physique plus rigoureuse est invité à se référer à l'annexe A.

<sup>2</sup>L'échelle mésoscopique se situe entre l'échelle macroscopique et l'échelle microscopique.

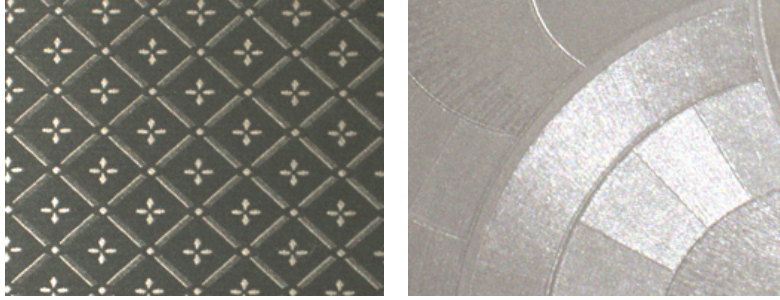


FIG. 2.1 – Exemple de matériaux représentables par une SVBRDF. Images tirées de [LBAD<sup>+</sup>06].



FIG. 2.2 – Un exemple de matériau représentable par une BTF. Images tirées de [MMS<sup>+</sup>06].

situés à l'échelle mésoscopique sont capturés par la BTF (*Bidirectional Texture Function*). Comme le montre la figure 2.2 les détails individuels tels que les plis ou les ombres générés par le matériau sur lui-même sont visibles à l'échelle mésoscopique.

La BRDF, la SVBRDF et la BTF modélisent des matériaux opaques et ne sont donc pas capables de représenter des phénomènes tels que la transparence ou la réfraction. Il en va de même pour les phénomènes dépendants du temps ou de la longueur d'onde tels que la phosphorescence (réémission de lumière dans le temps), la fluorescence (réémission de la lumière sur une longueur d'onde différente de celle initiale) ou encore la diffraction (diffusion de l'onde par des points de l'objet). En infographie, pour prendre en compte ce type de phénomènes, des traitements particuliers ou des fonctions plus générales sont utilisés.

### 2.1.2 Définition mathématique

La BRDF étant une fonction locale à la surface de l'objet, elle est exprimée dans un repère local  $(x, B, T, N)$  où :

- $x$  le point de la surface sur laquelle la BRDF s'applique
- $T$  la tangente de la surface au point  $x$
- $N$  la normale au point  $x$
- $B$  la binormale obtenue par le produit vectoriel de la tangente avec la normale.

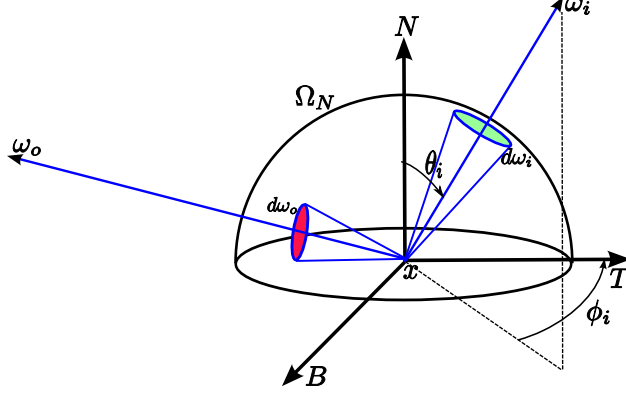


FIG. 2.3 – Repérage des directions pour la BRDF. La direction incidente  $\omega_i$  de la lumière et la direction d'observation  $\omega_o$  sont repérées dans le repère local  $(x, B, T, N)$  de la surface.  $\omega_i$  (resp.  $\omega_o$ ) substantive un angle solide différentiel  $d\omega_i$  (resp.  $d\omega_o$ ) sur l'hémisphère  $\Omega_N$ .

Exprimées dans ce repère, on note  $\omega_i$  la direction incidente de la lumière et  $\omega_o$  la direction d'observation (réfléchi) du point  $x$ . Ces directions sont exprimées sur l'hémisphère noté  $\Omega_N$  centré en  $x$  et orienté selon  $N$  (cf. figure 2.3).

On peut définir la BRDF, notée  $f_r$ , comme étant une fonction de ces deux directions et à valeurs non négatives :

$$f_r(\omega_i, \omega_o) : (\omega_i, \omega_o) \rightarrow \mathbb{R}_+. \quad (2.1)$$

Étant donné que la BRDF ne modélise que des matériaux opaques, seules les directions  $(\omega_i, \omega_o)$  contenues dans l'hémisphère orienté du côté de la normale sont pertinentes. On peut donc raffiner l'équation précédente pour obtenir le domaine de définition de la BRDF :

$$f_r(\omega_i, \omega_o) : \Omega_N \times \Omega_N \rightarrow \mathbb{R}_+. \quad (2.2)$$

En général, les directions  $(\omega_i, \omega_o)$  sont orientées de manière à avoir comme origine le point  $x$ .

### Paramétrisation

La manière classique d'exprimer les directions  $(\omega_i, \omega_o)$  est d'utiliser les coordonnées sphériques  $(\theta, \phi)$  où  $\theta$  représente l'angle colatitudinal et  $\phi$  l'angle azimutal (cf. figure 2.3) :

$$\begin{aligned} f_r(\theta_i, \phi_i; \theta_o, \phi_o) &: \Omega_N \times \Omega_N \rightarrow \mathbb{R}_+ \\ \theta_i, \theta_o &\in \left[0, \frac{\pi}{2}\right] \\ \phi_i, \phi_o &\in [0, 2\pi]. \end{aligned}$$

Exprimée de cette manière, la BRDF est une fonction de  $\mathbb{R}^4$  vers  $\mathbb{R}_+$ . La façon de repérer les directions sur l'hémisphère  $\Omega_N$  est appelée "paramétrisation de la BRDF" et constitue à elle seule un domaine de recherche. Les différentes formes de paramétrisation seront étudiées plus en détail à la section 2.2.2.

### Isotropie et anisotropie

Lorsque la BRDF est invariante par rotation autour de la normale, *i.e.*

$$\forall t, f_r(\theta_i, \phi_i; \theta_o, \phi_o) = f_r(\theta_i, \phi_i + t; \theta_o, \phi_o + t) \quad (2.3)$$

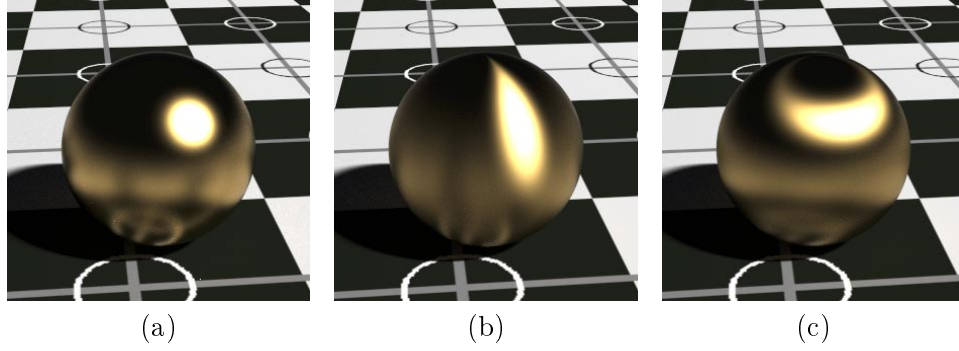


FIG. 2.4 – Illustration de la propriété d’isotropie et d’anisotropie d’une BRDF. (a) BRDF isotropique, BRDF anisotropique dont la direction principale d’anisotropie est (b) verticale ou (c) horizontale. Images tirées de [AS00].

et qu’elle est aussi invariante par réflexion dans le plan d’incidence, *i.e.*

$$f_r(\theta_i, 0; \theta_o, \phi_o) = f_r(\theta_i, 0; \theta_o, \pi - \phi_o) \quad (2.4)$$

on dit que la BRDF est isotropique ; dans le cas contraire, on parle de BRDF anisotropique. Des exemples de BRDFs isotropiques et anisotropiques sont montrés à la figure 2.4. Lorsqu’une BRDF est isotropique, elle ne dépend donc plus que de trois paramètres :  $f_r(\theta_i; \theta_o, \|\phi_o - \phi_i\|) : \mathbb{R}^3 \rightarrow \mathbb{R}_+$ .

Notons que pour une BRDF isotropique, la définition d’une tangente à la surface est inutile alors qu’elle est nécessaire pour une BRDF anisotropique.

### Contraintes physiques

L’ensemble des fonctions défini par l’équation 2.2 contient un sous-ensemble regroupant les BRDFs dites "physiquement réalistes". Ces BRDFs sont caractérisées par deux propriétés supplémentaires : la réciprocité et la conservation de l’énergie.

1. Réciprocité :

$$f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i) \quad (2.5)$$

2. Conservation de l’énergie :

$$\forall \omega_o, \int_{\Omega_N} f_r(\omega_i, \omega_o) \cos(N, \omega_i) d\omega_i \leq 1 \quad (2.6)$$

où  $d\omega_i$  est l’angle solide différentiel (cf. annexe A).

On utilise une inégalité pour représenter le fait que l’énergie lumineuse incidente peut être réfléchié ou absorbée sous forme de chaleur ou encore transmise (réfraction). Lorsque la totalité de l’énergie incidente est réfléchié, sans absorption ou transmission, l’inégalité précédente devient une égalité. La conservation de l’énergie est nécessaire pour s’assurer de la convergence des algorithmes d’éclairage global, tandis que la réciprocité facilite les techniques d’échantillonnage basé sur l’importance (*importance sampling*).



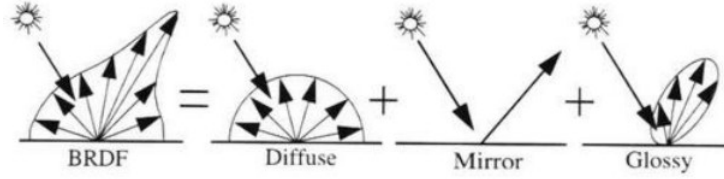


FIG. 2.5 – Construction d’une BRDF complexe à partir de BRDFs simples. De gauche à droite : La BRDF finale, la BRDF lambertienne, la BRDF spéculaire parfaite et la BRDF *glossy*.

### Longueur d’onde

En toute rigueur, la BRDF dépend de la longueur d’onde  $\lambda$  de la lumière incidente :

$$f_r(\omega_i, \omega_o, \lambda) : \Omega_N \times \Omega_N \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$$

et sa valeur n’est pas forcément égale pour chacune des longueurs d’onde. Ceci peut s’exprimer par :

$$f_r(\omega_i, \omega_o) = \int_{\lambda} f_r(\omega_i, \omega_o, \lambda) d\lambda.$$

En infographie, pour des raisons de stockage et de performance, on n’utilise que trois longueurs d’onde ( $\lambda_R, \lambda_B, \lambda_G$ ) pour représenter l’ensemble du spectre visible de la lumière. La BRDF s’exprime alors par :

$$f_r(\omega_i, \omega_o) = \alpha f_r(\omega_i, \omega_o, \lambda_R) + \beta f_r(\omega_i, \omega_o, \lambda_G) + \gamma f_r(\omega_i, \omega_o, \lambda_B) \quad (2.7)$$

où  $(\alpha, \beta, \gamma) \in \mathbb{R}_+^3$ . La BRDF modélise non seulement un filtre énergétique mais aussi un filtre colorimétrique. Dans la suite de ce document, on omettra la notation en  $\lambda$  lorsque l’aspect colorimétrique de la BRDF n’aura pas ou peu d’importance.

### BRDFs particulières

Il existe deux BRDFs particulières qui définissent les cas limites de comportement d’une BRDF. La BRDF uniforme ou lambertienne réfléchit la lumière de manière uniforme. Une définition analytique d’une BRDF lambertienne est donnée par :

$$f_r(\omega_i, \omega_o) = \frac{\rho_d}{\pi} \quad (2.8)$$

où  $\rho_d$ , albédo diffus, est une constante positive dépendante de la surface.

A l’opposé, la BRDF qui modélise le miroir parfait correspond à celle laquelle la lumière incidente est réfléchiée dans une et une seule direction :

$$f_r(\omega_i, \omega_o) = \delta \quad (2.9)$$

où  $\delta$  représente la fonction delta de Dirac étant uniquement non-nulle dans la direction  $\omega_r = (\theta_r, \phi_r) = (\theta_i, \pi + \phi_i)$ .

Ces deux BRDFs théoriques sont souvent utilisées conjointement pour définir de nouvelles BRDFs plus complexes comme illustré à la figure 2.5.

### 2.1.3 Aperçu de la recherche sur les BRDFs

#### Modèles analytiques

Les premiers travaux sur les BRDFs portaient sur le développement de modèles analytiques. On peut catégoriser les modèles suivant leur développement (empirique ou physique) ou encore selon qu'ils paramétrisent le lobe spéculaire autour de la direction  $\omega_r$  de la lumière réfléchie ou autour de la normale en utilisant le vecteur bissecteur  $H = \frac{\omega_o + \omega_i}{\|\omega_o + \omega_i\|}$ .

L'un des premiers modèles développés de manière empirique et qui reste encore largement employé est celui proposé par Phong [Pho75], dont la formulation est la suivante :

$$f_r(\omega_i, \omega_o) = \frac{\rho_d}{\pi} + \rho_s (\omega_r \cdot \omega_o)^k \quad (2.10)$$

avec :

- $\rho_d$  coefficient diffus
- $\rho_s$  coefficient spéculaire
- $k \in \mathbb{N}$  contrôle la taille du reflet spéculaire (*highlight*).

Le modèle de Phong est un exemple typique de création de BRDFs en utilisant le terme lambertien (composante diffuse) et une composante spéculaire allant de la réflexion miroir parfaite ( $k = \infty$ ) à la réflexion diffuse uniforme ( $k = 0$ ). Une alternative, moins coûteuse en terme de calcul et permettant de modéliser une BRDF anisotropique, a été proposée par Schlick [Sch94b] en utilisant des fonctions rationnelles pour calculer la composante spéculaire du modèle.

Le modèle de Phong n'étant ni réciproque et ne conservant pas l'énergie, il ne constitue pas une BRDF qui satisfait les contraintes de propriétés physiques. Lafortune et Willems [LW94] ont proposé un modèle de Phong alternatif qui satisfait les contraintes physiques grâce à une reformulation du terme  $\rho_s$ . Ils ont également proposé une généralisation de ce modèle [LFTG97], dont le principe est de construire une BRDF qui possède plusieurs lobes, pour pouvoir modéliser des comportements anisotropiques. D'autres modèles, n'utilisant pas forcément le vecteur  $\omega_r$ , de BRDFs anisotropiques avaient été proposés, notamment [Kaj85, PF90, Sch94a].

Le premier modèle à paramétriser le lobe spéculaire en utilisant le bissecteur  $H$  a été introduit par Blinn [Bli77] suite aux travaux de Torrance et Sparrow [TS67]. Sa formulation est très proche de celle proposée par Phong :

$$f_r(\omega_i, \omega_o) = \frac{\rho_d}{\pi} + \rho_s (N \cdot H)^k. \quad (2.11)$$

Ward [War92] a dérivé de manière empirique un modèle où l'anisotropie est contrôlée par deux paramètres définissant une ellipse et qui utilise également le vecteur  $H$ .

Une observation faite par certains chercheurs montre que les coefficients  $\rho_d$  et  $\rho_s$  ne sont pas constants en général et dépendent de l'angle d'incidence de la direction d'observation. A savoir, plus on observe à angle rasant certaines surfaces, plus le coefficient spéculaire devient important, tandis que le coefficient diffus diminue de manière à ce que la conservation de l'énergie soit respectée. Ce phénomène est appelé effet de Fresnel et plusieurs modèles analytiques [Sch94a, LFTG97, AS00, APS00] en tiennent compte. La figure 2.6 montre un exemple de matériau pour lequel l'effet de Fresnel est visuellement important.

Enfin signalons que le terme diffus a lui aussi été modélisé de manière plus précise par Oren et Nayar [ON94] qui montrent que pour certains matériaux diffus plus rugueux, le modèle lambertien est inadéquat.

Parallèlement à la recherche sur les modèles empiriques, des modèles physiques ont également été développés tels que celui de Cook et Torrance [CT82], celui de He et al. [HTSG91], et plus

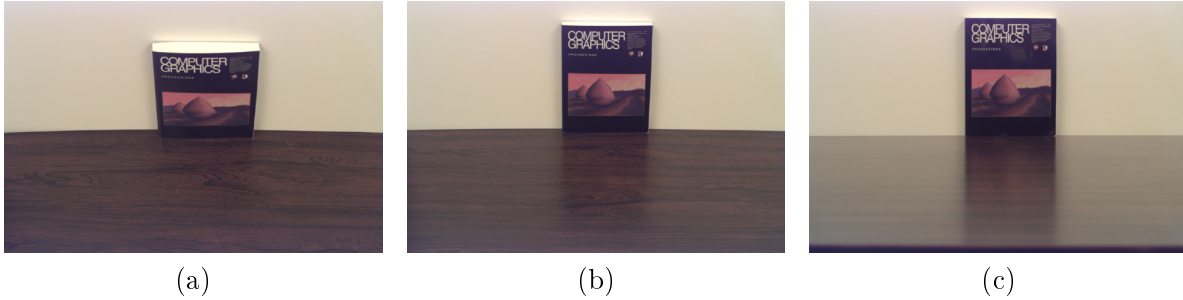


FIG. 2.6 – Illustration de l’effet de Fresnel sur un matériau comme le bois. (a) La table en bois vue à incidence normale, le coefficient diffus domine le coefficient spéculaire. (b) Pour une incidence à 45 degrés, les coefficients sont plus ou moins égaux. (c) A incidence rasante le coefficient spéculaire domine. Photographies tirées de [LFTG97].

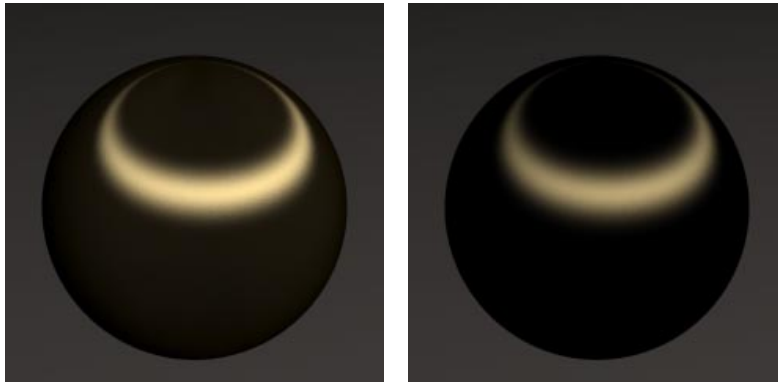


FIG. 2.7 – Comparaison entre le modèle de Ward [War92] et le modèle d’Ashikhmin et al. [APS00]. Images tirées de [APS00].

récemment celui d’Ashikhmin et al. [APS00]. La figure 2.7 compare l’apparence obtenue sur une sphère en utilisant le modèle de Ward [War92] et le modèle d’Ashikhmin et al. [APS00].

Les modèles physiques s’assurent nécessairement de respecter les propriétés de réciprocité et de conservation de l’énergie. Le modèle proposé par He et al. tient aussi compte de la polarisation de la lumière et est certainement le modèle le plus complet. En pratique il reste peu utilisé parce qu’il contient de trop nombreux paramètres peu intuitifs à contrôler pour un utilisateur non expert.

Sans entrer dans le détail de chacun des modèles physiques, il convient de rappeler l’approche prise pour modéliser la BRDF. A l’exception du modèle de He et al., les modèles physiques utilisent le concept de distribution de microfacettes pour caractériser la surface de l’objet à l’échelle microscopique. Une microfacette est une surface plane et orientée qui se comporte comme un miroir parfait. Une fonction de distribution des microfacettes donne la proportion de microfacettes qui entrent en jeu pour le calcul de la BRDF en fonction des directions de vue et de lumière. Il faut tenir compte du masquage (ou ombrage) des facettes à l’aide d’un terme supplémentaire dans le modèle, soit un coefficient d’atténuation géométrique ou un terme d’ombrage. La figure 2.8 illustre le concept de terme d’ombrage. Le plus souvent, on factorise la dépendance sur la direction de vue et de lumière en utilisant uniquement le vecteur  $H$  comme seul paramètre du terme d’ombrage et

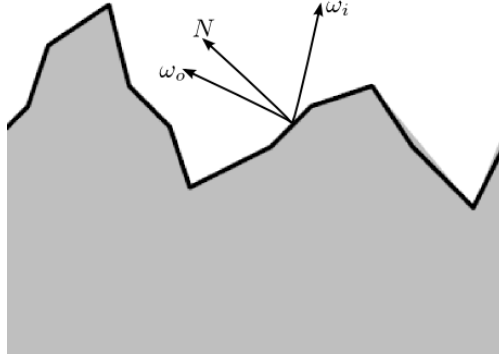


FIG. 2.8 – L’effet d’ombrage des microfacettes. En fonction de la direction de vue ( $\omega_o$ ) et de la direction de la lumière ( $\omega_i$ ), une proportion des microfacettes ne sont pas actives : elles sont masquées par d’autres microfacettes.

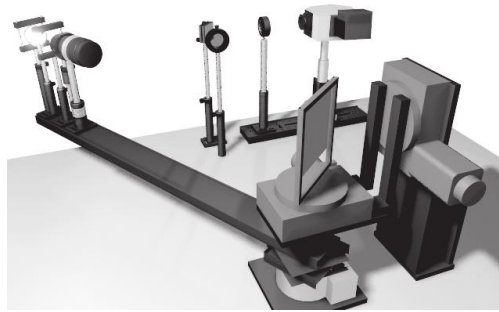


FIG. 2.9 – Exemple de gonioréfectomètre. Image tirée de [LFTW05].

de la distribution de microfacettes.

En résumé, les deux approches de développement de modèles analytiques sont intéressantes et chaque type de modèle a sa place. Lorsqu’on souhaite modéliser un matériau où la dispersion de la lumière (*light scattering*) est négligeable, les modèles empiriques sont suffisants. Si la dispersion est une donnée importante, les modèles physiques sont plus appropriés.

### Mesure et *fitting* de BRDF

Une voie explorée pour construire des BRDFs, récemment redevenue populaire, est de mesurer directement la réflectance des matériaux à l’aide de dispositifs spécialisés [CMS87, WAT92, War92, MPBM03]. Le plus classique est le gonioréfectomètre [CMS87] qui permet de faire varier la position de la source de lumière et celle d’un capteur pour acquérir des valeurs discrètes de la BRDF. La figure 2.9 montre un exemple récent de gonioréfectomètre. Outre le fait que l’acquisition des données de réflectance est un procédé long et fastidieux, l’un des principaux problèmes est la robustesse des mesures acquises pour les angles rasants de la surface ou encore pour les configurations où la rétro-réflexion<sup>3</sup> entre en jeu. Dernièrement, Matusik et al. [MPBM03] ont mesuré une grande quantité de BRDFs isotropiques. Conjointement avec Ngan et al. [NDM05], un nouveau système de mesure a été développé pour mieux acquérir les BRDFs anisotropiques.

<sup>3</sup>On parle de rétro-réflexion lorsque la lumière se réfléchit dans la direction d’où elle provient.

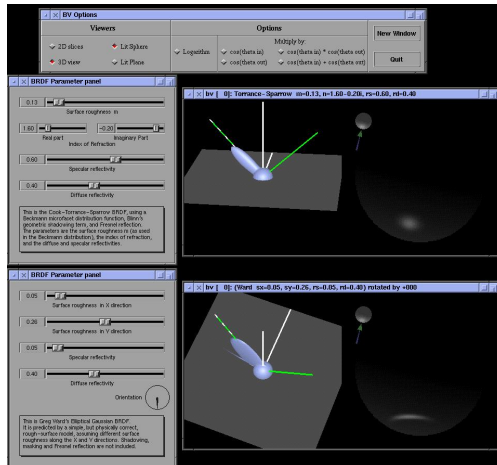


FIG. 2.10 – **bv** le logiciel de visualisation de BRDFs développé par Rusinkiewicz.

En général, la taille des données acquises est trop volumineuse pour être directement exploitables. Une manière d’exploiter les données acquises est d’essayer, par un mécanisme de minimisation d’erreur, de trouver un modèle analytique qui approxime les données mesurées [War92, LFTG97, MPBM03, NDM05, CBP07]. Dans une récente étude, Ngan et al. [NDM05] ont montré que le modèle d’Ashikhmin et al. [APS00] et la paramétrisation du lobe spéculaire utilisant le bissecteur  $H$  possède de meilleures propriétés de convergence pour les algorithmes de *fitting*. Ceci est notamment dû au fait que le vecteur  $H$  est parfaitement défini lorsque les vecteurs  $\omega_o$  et  $\omega_i$  sont rasants à la surface, là où justement la robustesse des données acquises est faible. Une autre conclusion tirée par Ngan et al. est que pour les matériaux isotropiques, les modèles analytiques de BRDFs réussissent bien à modéliser l’apparence du phénomène alors que pour les BRDFs anisotropiques, ils échouent dans de nombreux cas à bien représenter l’apparence du matériau.

## Visualisation des BRDFs

Nous terminons cet aperçu en mentionnant des travaux qui mettent un peu plus l’emphase sur l’utilisateur. Nous détaillerons à la section 2.1.4 les différents systèmes d’édition. Avec quatre dimensions, la BRDF est une fonction complexe à visualiser. Afin de faciliter cette visualisation et le développement d’une intuition sur le comportement de la BRDF, des logiciels de visualisation ont été développés. En général, la direction de vue ou de lumière est fixée afin d’obtenir une représentation tridimensionnelle des valeurs de la BRDF. Rusinkiewicz a développé un logiciel de visualisation (<http://graphics.stanford.edu/~smr/brdf/bv/>) dont une capture d’écran est montrée en figure 2.10. Heckbert en propose un également (<http://www.cs.cmu.edu/~ph/src/illum/>). Ces logiciels permettent de visualiser des BRDFs analytiques.

Ce premier aperçu de la recherche sur les BRDFs nous montre que peu d’emphase a été mis sur le contrôle ou l’édition de la BRDF par l’utilisateur. Souvent les modèles analytiques (à quelques exceptions près) possèdent des paramètres complexes et difficiles à assimiler par un artiste. Dans cette proposition, nous souhaitons réintroduire l’artiste et l’utilisateur novice dans le processus de création de BRDFs en lui proposant deux paradigmes d’édition qui lui permettront de construire et dessiner sa propre BRDF. A savoir, nous plaçons l’emphase sur la spécification, par l’artiste, de l’apparence finale de la surface de l’objet plutôt que sur la définition ou le contrôle direct des paramètres de la BRDF. Nous choisissons de nous concentrer sur l’édition de la composante spéculaire

(lobe) de la BRDF parce qu'elle définit intrinsèquement ce qui est le plus apparent dans une BRDF. Dans notre approche la composante diffuse reste contrôlable mais ne présente pas de réelle difficulté de contrôle ou d'édition.

Avant de présenter notre approche, nous passons en revue les différentes approches et systèmes qui existent pour éditer et contrôler les BRDFs.

#### 2.1.4 Travaux antérieurs sur l'édition interactive de BRDF

Les travaux d'édition et de contrôle de BRDFs peuvent être divisés en deux catégories. Ceux portant sur l'édition interactive de modèles analytiques dans des conditions particulières et ceux mettant en oeuvre un processus inverse pour répondre aux besoins de l'artiste.

##### Édition de modèles analytiques/paramétriques

Les modèles paramétriques étant le plus souvent difficiles d'approche, un certain nombre de systèmes interactifs ont été proposés afin de permettre à l'utilisateur d'apprendre à mieux cerner l'influence des paramètres du modèle. L'exemple le plus probant est le système proposé par He et al. [HHP<sup>+</sup>92], sensé donner une intuition sur l'action des paramètres du modèle de BRDFs proposé par les mêmes auteurs l'année précédente. Le système de Ben-Artzi et al. [BAOR06] propose à l'utilisateur d'éditer la BRDF d'un objet statique éclairé par un *environment map*, à la différence de la visualisation de Rusinkiewicz qui ne propose de l'édition qu'avec des lumières ponctuelles ou directionnelles. Une représentation de la BRDF à base de courbes 1D (cf. figure 2.11) permet à l'utilisateur de manipuler les paramètres de la BRDF. En fait, l'utilisateur ne change pas les paramètres du modèle analytique sous-jacent mais une représentation de ces paramètres et le système se charge alors de la transformation. Cela permet une manipulation linéaire de la BRDF. Le même système d'édition de BRDF est utilisé dans l'*Inverse Shade Tree* de Lawrence et al. [LBAD<sup>+</sup>06]. Ben-Artzi et al. ont proposé une extension [AKFR07] de leur système afin d'éditer dans un contexte de lumière encore plus complexe comme les scènes d'intérieur utilisées en éclairage global. Enfin, signalons que le modèle proposé par Ashikhmin et al. [APS00], permet une certaine forme limitée d'édition. La distribution des microfacettes peut être spécifiée par l'artiste en utilisant une simple texture 2D et permet d'obtenir des lobes spéculaires de tout type (cf. figure 2.12). Malheureusement, la flexibilité laissée à l'utilisateur est limitée puisqu'il ne peut effectuer comme seule action que choisir sa texture qui est 2D pour une spécifier la BRDF qui est 4D. De plus, il nous semble que, dès que la direction de vue change, le comportement du lobe spéculaire risque de ne plus correspondre à la texture fournie et pourrait devenir incohérent.

##### Edition par processus inverse

De manière générale en rendu inverse, l'utilisateur spécifie le résultat qu'il souhaite obtenir (spécification des contraintes) et le système tente de les satisfaire. Un des premiers systèmes introduit par Poulin et Fournier [PF92] et amélioré dans [PRJ97] laisse l'utilisateur dessiner les *highlights* et les ombres et le système se charge de modifier les lumières en conséquence. Plus en rapport avec les BRDFs le système proposé dans [PF95] laisse l'utilisateur peindre des couleurs sur la surface et le système via une minimisation non linéaire calcule les paramètres du modèle de Phong qui correspondent au mieux au dessin de l'utilisateur. Reprenant la même approche, Colbert et al. [CPK06] ont récemment proposé de laisser l'utilisateur peindre les *highlights* d'une BRDF utilisant le modèle analytique de Ward. Un processus inverse construit un lobe tel que défini par la fonction de Ward [War92]. Afin de permettre plus de liberté de création, la BRDF n'est pas contrainte à un

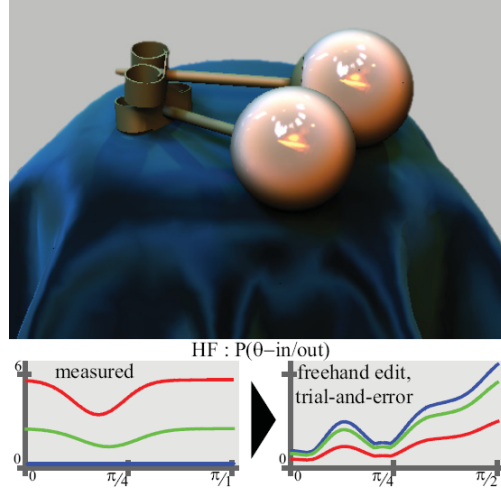


FIG. 2.11 – Illustration du système d’édition, à l’aide de courbes, de BRDFs de Ben-Artzi et al. Image tirée [BAOR06].

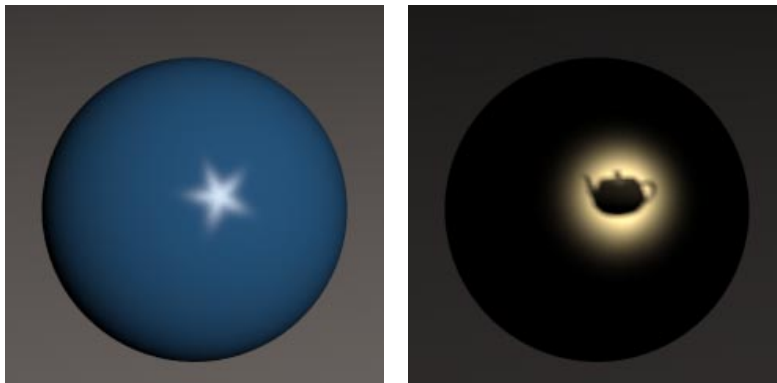


FIG. 2.12 – Utilisation d’une texture pour définir la distribution des microfacettes du modèle d’Ashikhmin et al.[APS00].

seul lobe mais à une somme de lobes qui respectent les contraintes de conservation d’énergie et de réciprocité. L’intérêt d’avoir une BRDF composée de plusieurs lobes est d’augmenter les possibilités de création artistique. Une capture d’écran du système (**BRDF Shop**) proposé par Colbert et al. est présentée à la figure 2.13. Contrairement à ce qui est fait dans [PF95], l’absence de processus de minimisation de l’erreur au carré permet à **BRDF Shop** d’être un système interactif.

L’une des principales limitations du système précédent est son manque de liberté dans les outils de dessin proposés. Ceci est dû à l’approche choisie, à savoir résoudre un problème inverse. Pour s’assurer que le problème inverse soit toujours résoluble, les outils d’édition sont limités à des fonctions inversibles, comme un filtre gaussien. Il en résulte que les formes de *highlights* dessinables sont peu variées. De plus, bien que l’artiste voit le résultat de son dessin sur l’objet qui l’intéresse, il ne peint pas directement dessus mais sur une sphère affichée en même temps.

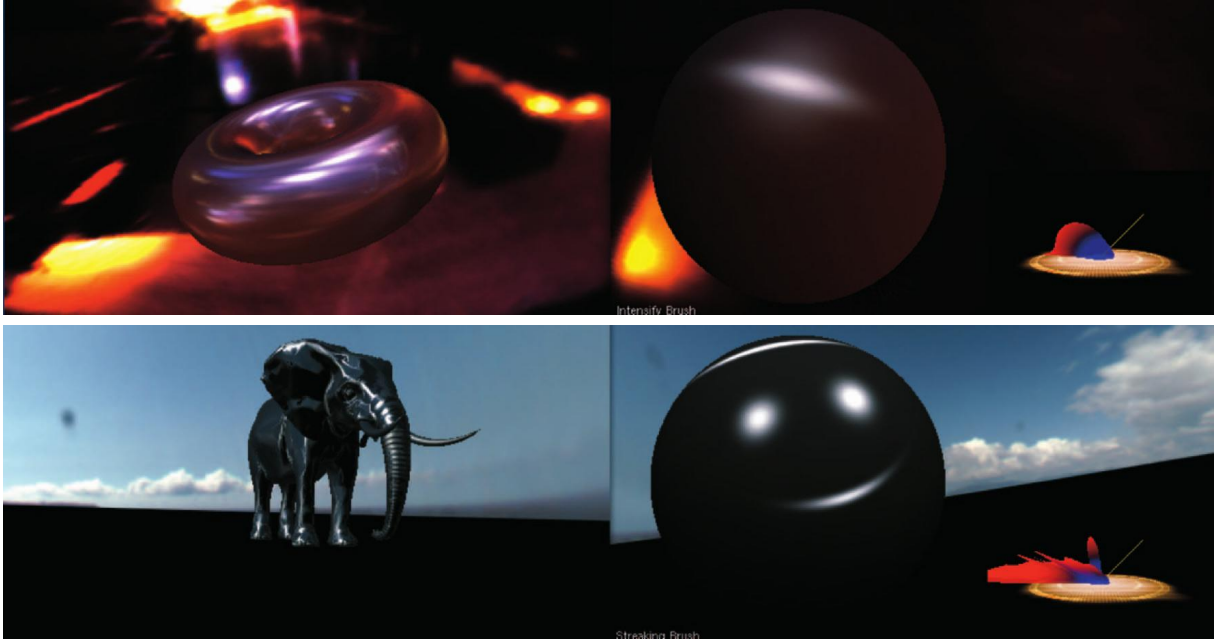


FIG. 2.13 – **BRDF Shop**. Images de gauche : le résultat de l'édition de la BRDF sur un objet. Images de droite : la sphère que l'utilisateur édite pour créer sa BRDF représentée en bas à droite. En haut, une édition de BRDF avec un seul lobe. En bas, édition d'une BRDF avec trois lobes. Images tirées de [CPK06].

### Autres formes d'édition

D'autres systèmes d'édition dans le cadre du rendu non-photoréaliste existent, *Lit Sphere* proposé par Sloan et al. [SMGG01] où l'utilisateur peint également sur une sphère. Les données acquises sur la sphère sont ensuite reprojétées sur la surface 3D en prenant en compte les similarités entre le point de vue et les normales, limitant ainsi le système à une scène statique. Enfin, signalons deux systèmes qui proposent un paradigme de peinture sans pour autant travailler sur la BRDF. Celui proposé par Anjyo et al. [AWB06] laisse l'utilisateur peindre des ombres ou des reflets de manière artistique et en déduit des formes de lumière. Celui proposé par Okabe et al. [OMS<sup>+</sup>07] déduit la valeur d'un *Environment Map* en fonction de la couleur apposée sur la surface de l'objet par l'utilisateur.

Force est de constater que la problématique de l'édition des BRDFs a été peu abordée comparativement à celle du développement de modèles analytiques ou de la mesure. Les systèmes d'édition de modèles paramétriques limitent la créativité de l'utilisateur aux formes de lobes spéculaires définies par le modèle. Cependant, la précision d'édition proposée par le changement de courbes ou de valeurs est indéniable. Les systèmes inverses, qui permettent de peindre, présentent une interface d'édition qui semble nettement plus conviviale, mais restent eux aussi limités soit par la résolution inverse du problème, soit dans la diversité de création de formes. Les deux formes d'édition présentent des avantages et des inconvénients et le paradigme idéal se situe probablement à mi-chemin.



## 2.2 Recherche proposée en édition

### 2.2.1 Motivation

Notre approche vise à donner plus de flexibilité à l'artiste pour la création des *highlights*. Il ne s'agit pas d'obtenir à tout prix une BRDF physiquement réaliste. En effet, les formes de *highlights* dessinables par l'artiste ne respecteront pas nécessairement toute la complexité de la réalité physique. Nous allons donc relâcher les contraintes imposées par certaines lois physiques telles que la conservation de l'énergie et la réciprocité. Cependant, il nous apparaît important que la BRDF modélisée garde un comportement "cohérent". On entend ici par cohérent le fait qu'un déplacement de la direction de vue ou de lumière affecte la forme et l'intensité du *highlight* avec une certaine continuité attendue. Cette cohérence dans le comportement guidera nos algorithmes pour conserver certains résultats macroscopiques sur la taille, la concentration ou encore la couleur du *highlight* tant qu'ils ne restreignent pas la liberté d'édition.

Dans ce document, nous proposons deux paradigmes d'édition pour les BRDFs, l'un orienté édition directe tandis que l'autre est orienté édition inverse. Dans chacun des paradigmes l'édition du lobe spéculaire se fait sous une source de lumière à la fois, de même que pour une direction de vue à la fois. Le coefficient diffus est supposé uniforme sur toute la surface, mais reste éditable. Afin d'éditer la BRDF, une représentation efficace doit être fournie, et cette représentation nécessite de choisir une paramétrisation adéquate (cf. section 2.2.2). Enfin, nos deux paradigmes permettent une édition incrémentale au sein de laquelle l'utilisateur peut raffiner sa BRDF.

Le premier paradigme d'édition est un paradigme d'édition directe orienté dessin vectoriel. L'utilisateur contrôle la forme et le gradient du lobe à l'aide de formes paramétrées. Il édite la forme dans un plan 2D perpendiculaire à la direction du vecteur lumière (ou réfléchi). Ce paradigme d'édition permet une édition intuitive et précise. De plus, l'utilisation de formes vectorielles assure une continuité de la représentation. En effet, comme nous le verrons à la section 2.2.2, nous définissons les formes de manière à garantir une interpolation continue lorsque les directions de lumière ou d'observation sont modifiées. Grâce à ce schéma d'interpolation, un nombre peu élevé (2 ou 3) de dessins doit être fourni par l'utilisateur pour obtenir une BRDF opérationnelle. En revanche, cette contrainte de continuité impose une restriction sur le type de formes éditables. Ce premier paradigme s'apparente à l'édition directe de courbes proposée par Ben-Artz et al. [BAOR06] avec comme différence majeure que les courbes éditées contrôlent de manière directe la forme du lobe et non des paramètres d'une BRDF qui ont parfois un effet moins évident sur l'apparence dans l'image.

Dans la tradition de l'édition inverse orientée peinture, notre second paradigme propose à l'utilisateur de peindre son lobe directement sur la surface de l'objet. Il s'agit donc d'un paradigme d'édition très général dans lequel la géométrie de la surface est directement prise en compte lors du processus de création de la BRDF. Il permet également de "plaquer" des *highlights* pris à partir d'une photo, à condition que l'utilisateur fournisse une direction de lumière et d'observation. La principale inconnue dans ce paradigme est que le comportement cohérent, tel que défini par le système, ne soit pas celui attendu par l'utilisateur. Bien que ce paradigme puisse sembler plus fastidieux pour qu'un utilisateur aboutisse rapidement à une BRDF opérationnelle, nous proposons également des pistes qui permettront d'accélérer le processus de création.

### 2.2.2 Paramétrisations de la BRDF

Dans cette section nous donnons plus de détails sur les différentes paramétrisations d'une BRDF. La paramétrisation est une notion importante parce qu'elle possède une influence sur l'utilisation

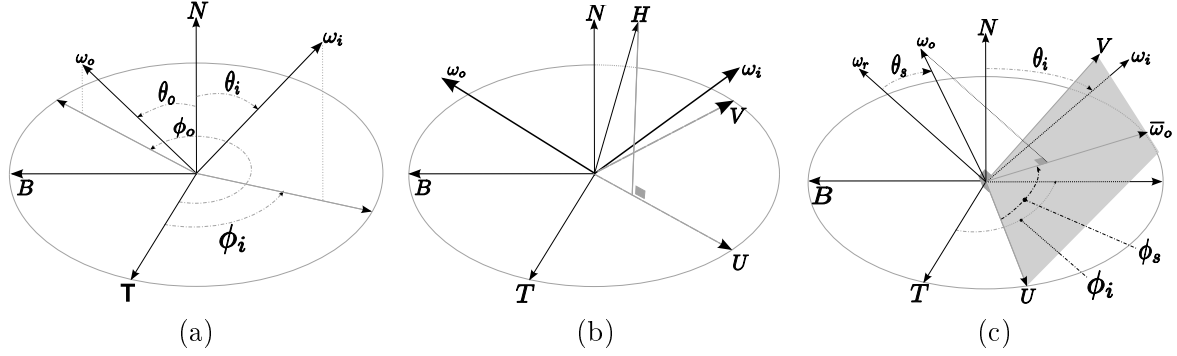


FIG. 2.14 – Les différents types de paramétrisation. (a) Classique, (b) Rusinkiewicz [Rus98], (c) la nôtre.

dont on veut faire de la BRDF. Une paramétrisation spécifique est souvent nécessaire pour chaque type d'utilisation de la BRDF. Par exemple en *fitting*, la paramétrisation de Rusinkiewicz [Rus98] donne les meilleurs résultats, alors que pour du rendu sur GPU, celles de Kautz et McCool [KM99a, MAA01] sont plus appropriées. Dans cette section, nous mettons en évidence les lacunes de chaque paramétrisation. Nous proposons une nouvelle paramétrisation qui possède de bonnes propriétés pour les paradigmes d'édition que nous proposons.

Rappelons que sauf mention contraire, les directions sont exprimées de manière locale et qu'elles ont été transformées au préalable du repère global (scène) dans le repère local (surface). La figure 2.14 regroupe les schémas illustratifs des différentes paramétrisations que nous allons aborder.

### Paramétrisation classique

La manière classique de repérer les vecteurs  $\omega_i$  et  $\omega_r$  est d'utiliser les coordonnées sphériques  $(\theta, \phi)$  dans le repère local de la surface. La transformation classique de coordonnées cartésiennes en coordonnées sphériques est donnée à l'équation 2.12.

Une direction  $\omega = (x, y, z)$  normalisée et repérée sur l'hémisphère  $\Omega_N$  peut s'exprimer en coordonnées sphériques  $(\theta, \phi)$  en utilisant le système d'équations suivant :

$$\begin{cases} \theta = \arccos z \\ \phi = \arccos(\bar{\omega} \cdot T) \\ \theta \in [0, \frac{\pi}{2}] \quad \phi \in [0, 2\pi] \end{cases} \quad (2.12)$$

où :

- dans le repère local  $(B, T, N)$ , une direction  $\omega = x \cdot B + y \cdot T + z \cdot N$
- $\bar{\omega}$  est la projection de  $\omega$  dans le plan formé par les vecteurs  $B$  et  $T$ .

Les équations 2.12 mettent en évidence les problèmes de continuité qui surviennent lorsqu'on utilise la paramétrisation classique. En effet, l'angle  $\phi$  n'est plus défini lorsqu'une direction est confondue (ou numériquement proche) avec la normale de la surface, puisque  $\bar{\omega}$  devient le vecteur nul. En revanche, l'angle  $\theta$  est toujours défini. Ceci justifie que pour tous les modèles analytiques isotropiques qui ne dépendent pas de  $\phi$ , aucun problème ne se pose en pratique.

### Paramétrisation de Rusinkiewicz

La paramétrisation de Rusinkiewicz [Rus98] est illustrée à la figure 2.14(b). Elle utilise le vecteur

bissecteur  $H$  et un vecteur différence  $D$  définis comme suit :

$$\left\{ H = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|} \right. \quad (2.13)$$

et  $D$  est défini à l'aide de vecteurs  $U$  et  $V$  par :

$$\left\{ \begin{aligned} U &= -\frac{(N - (N \cdot H)H)}{\|(N - (N \cdot H)H)\|} \\ V &= H \times U \\ D &= [\omega_i \cdot H, \omega_i \cdot U, \omega_i \cdot V]. \end{aligned} \right. \quad (2.14)$$

Dans cette paramétrisation, lorsque  $H \equiv N$ , une discontinuité apparaît puisque  $U$  devient le vecteur nul de même que  $V$ . Il faut bien noter que  $D \equiv \omega_i$ . En fait, il s'agit de projeter  $\omega_i$  dans le repère  $(H, U, V)$ .

### Paramétrisation de Kautz et McCool

La paramétrisation proposée par Kautz et McCool [KM99b] est très proche de celle de Rusinkiewicz à la différence que la discontinuité apparaît lorsque  $H$  est rasant à la surface. La seule différence se situe au niveau de la définition de  $U$  et  $V$  :

$$\left\{ \begin{aligned} U &= -\frac{(T - (T \cdot H)H)}{\|(T - (T \cdot H)H)\|} \\ V &= H \times U. \end{aligned} \right. \quad (2.15)$$

La discontinuité apparaît désormais lorsque  $H$  est colinéaire à  $T$ , ce qui arrive dans le cas de la rétro-réflexion. Il n'y a donc aucune différence fondamentale entre la paramétrisation de Kautz et McCool [KM99b] et celle de Rusinkiewicz [Rus98] si ce n'est la construction du repère  $(H, U, V)$  qui ne donne pas lieu au même cas de discontinuité. Kautz et McCool justifient leur paramétrisation en rappelant que les cas de rétro-réflexion sont moins fréquents que ceux où  $H = N$ .

Cependant, la discontinuité de cette paramétrisation constitue, selon nous, une limitation dans un contexte d'édition. L'utilisateur pourrait très bien vouloir spécifier un lobe pour le cas rétro-réflexif. Remarquons que les panneaux de signalisation routière constituent un exemple simple de rétro-réflexion.

### Notre paramétrisation

La paramétrisation que nous proposons possède des similarités avec celle esquissée par Kautz et McCool [KM00]. La figure 2.14(d) donne un schéma explicatif de notre paramétrisation. Comme indiqué sur la figure 2.14, nous préférons utiliser  $\omega_r$  plutôt que le vecteur  $H$  pour paramétriser le lobe. En effet, le vecteur  $H$  dépend à la fois des directions de vue et de lumière, et n'est donc pas approprié pour une édition à direction de lumière fixe. Nous paramétrisons  $\omega_o$  en utilisant  $(\theta_s, \phi_s)$  tels que définis ci-dessous :

$$\left\{ \begin{aligned} \cos \theta_s &= \omega_o \cdot \omega_r \\ \cos \phi_s &= \omega_o \cdot U \\ \sin \phi_s &= \omega_o \cdot V \end{aligned} \right.$$

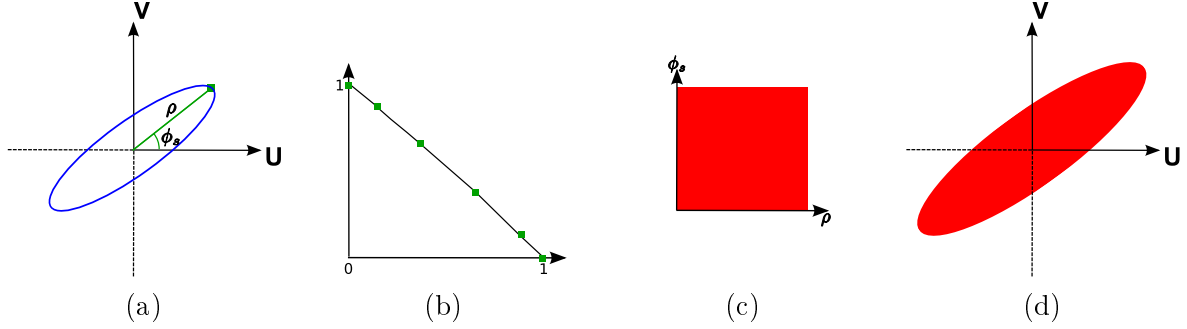


FIG. 2.15 – Paradigme d'édition vectorielle. Principe de l'édition du lobe. Les points de contrôle sont affichés en vert. (a) Définition par l'utilisateur de la courbe  $\mathcal{C}(\phi_s)$  qui spécifie la forme du lobe. (b) Définition de la fonction de potentiel  $\mathcal{P}$ , ici choisie de manière linéaire. (c) Choix de la couleur qui peut être fonction de  $\rho$  et  $\phi_s$ . (d) Lobe final obtenu.

avec  $U$  and  $V$  définis par :

$$\begin{cases} V = \frac{N - (N \cdot \omega_r) \omega_r}{\sqrt{1 - (N \cdot \omega_i)^2}} \\ U = V \times \omega_r. \end{cases}$$

Intuitivement,  $(\theta_s, \phi_s)$  paramétrisent une calotte polaire où  $\omega_r$  est l'axe du pôle nord. Notons que la seule discontinuité de cette paramétrisation survient lorsque  $\omega_i$  est confondu avec la normale. En effet, le produit scalaire  $N \cdot \omega_i = 1$  et par conséquent le vecteur  $V$  n'est plus défini. Il en va de même, par construction, pour le vecteur  $U$ . Remarquons que les vecteurs  $U$  et  $V$  sont définis même lorsque  $\omega_i$  et/ou  $\omega_r$  sont rasants à la surface. En effet, dans le cas limite où  $\omega_i$  est colinéaire avec  $T$ ,  $V$  est défini et vaut  $V = N$  ainsi que  $U$  qui vaut  $U = B$ .

### 2.2.3 Paradigme d'édition orienté dessin vectoriel

Dans cette section nous décrivons notre premier paradigme d'édition. Nous détaillons le processus d'édition du point de vue de l'utilisateur avant de préciser comment le lobe est représenté à l'aide de notre paramétrisation. Nous précisons comment la variation du lobe, en fonction d'un déplacement de la direction de vue ou de lumière, est contrôlée avant de présenter quelques pistes d'implémentation.

#### Édition

Le type de source de lumière est soit directionnel ou ponctuel. La figure 2.15 illustre les différents paramètres d'édition fournis à l'utilisateur : la forme, le gradient et la couleur du lobe. Pour une direction de lumière fixée, l'utilisateur édite le lobe dans un plan perpendiculaire à la direction de lumière  $\omega_i$  (rétro-réflexion) ou à  $\omega_r$  (réflexion). Ce plan est défini par les vecteurs  $U$  et  $V$  de la paramétrisation présentée à la section précédente. Le lobe dessiné par l'utilisateur dans ce plan, qui est donc fonction de la direction de lumière, constitue ce que nous appelons une configuration vue-lumière. Le paragraphe suivant détaille comment le système interpole entre les différentes configurations.

L'édition dans le plan  $UV$  se fait à l'aide d'une courbe paramétrée polairement qui définit la forme du lobe (cf. figure 2.15(a)) et d'une fonction de potentiel (figure 2.15(b)) paramétrée radialement qui

définit le gradient de l'énergie du lobe. La courbe paramétrée et la fonction de potentiel définissent un champ d'iso-courbes dans le plan  $UV$  dont un exemple est donné en figure 2.15(d). Enfin, la couleur de champ est également paramétrable (figure 2.15(c)).

### Représentation

Nous donnons désormais une définition mathématique aux différents paramètres présentés en figure 2.15 qui permettent d'éditer le lobe. La forme du lobe est définie par une courbe polaire qui repose dans le plan formé des vecteurs  $U$  et  $V$ . On note  $\mathcal{C}$  la courbe polaire contrôlant la forme du lobe :

$$\mathcal{C}(\phi_s) = \rho \quad (2.16)$$

où  $\rho$ , le rayon de la courbe, est défini par :  $\rho = 1 - \cos \theta_s$ . On pourrait utiliser une autre formulation pour définir  $\rho$  voire laisser l'utilisateur modifier ce paramètre. Cependant, le contrôle de ce paramètre ne semble pas intuitif, à première vue. Notons que la formulation proposée ci-dessus pour  $\rho$  donne plus de précision (variation faible du cosinus) lorsqu'on se trouve au centre du lobe (valeurs faibles de  $\theta_s$ ), ce qui présente un avantage.

Plutôt que de nous restreindre à des courbes polaires analytiques, nous utilisons des *spline-based polar curves* telles qu'introduites par Crespín et al. [CBS96], qui permettent d'obtenir des formes anisotropiques (cf. figure 2.15(a)). L'utilisateur définit non seulement l'axe principal de l'anisotropie mais aussi les différents points de contrôle de la courbe. Comme il s'agit d'une paramétrisation polaire, les points de contrôle sont au départ placés sur un cercle et l'utilisateur les déplace le long de leur rayon. Le choix du type d'interpolation pour la courbe (linéaire, cubique ou plus) est un paramètre utilisateur.

L'intensité  $I(\rho, \phi_s)$  du lobe est définie par :

$$\begin{cases} I(\rho, \phi_s) = \mathcal{P}(t) \\ t = \frac{\rho}{\mathcal{C}(\phi_s)}. \end{cases} \quad (2.17)$$

Nous appelons  $\mathcal{P}$  la fonction de potentiel. Dans le modèle de Phong, cette fonction est monotone décroissante et vaut :

$$\mathcal{P}(t) = (1 - t)^n \quad \text{avec } t \in [0, 1] \quad \text{et } n \in \mathbb{R}_+^* . \quad (2.18)$$

Pour plus de flexibilité nous laissons l'utilisateur éditer  $\mathcal{P}$  qui prend la forme d'une B-Spline. Ceci l'autorise à créer une courbe qui possède des minima et maxima locaux. Comme  $t$  dépend de  $\mathcal{C}$ , la fonction  $\mathcal{C}$  couplée à la fonction  $I(\rho, \phi_s)$  forment un champ de potentiel reposant sur des iso-courbes définies par la courbe  $\mathcal{C}$ . La valeur de chaque iso-courbe est contrôlée par la fonction  $\mathcal{P}(t)$ .

La couleur du lobe peut être fixée par l'utilisateur de manière indépendante de  $\rho$  et de  $\phi_s$ , mais elle peut aussi être paramétrée en fonction de  $\rho$  et/ou de  $\phi_s$ . L'apparence finale du lobe est donc obtenue en multipliant la fonction de couleur par la fonction d'intensité  $I(\rho, \phi_s)$ .

### Configurations

Avant de décrire le système d'interpolation, nous décrivons le nombre de configurations vue-lumière que l'utilisateur doit définir pour obtenir une BRDF opérationnelle.

Pour une BRDF isotropique, deux configurations sont nécessaires :

1. lorsque l'angle entre la direction de la lumière et la normale est nul

2. lorsque l'angle entre la direction de la lumière et la normale est de 45 degrés.

Le système interpole ensuite la forme du lobe entre ces deux configurations. Pour les angles supérieurs à 45 degrés, la configuration utilisée est celle à 45 degrés. Habituellement une BRDF isotropique physiquement réaliste possède un lobe en forme de disque pour la première configuration. Nous proposons dans notre système de générer par défaut un disque, mais l'utilisateur peut décider de changer la forme du lobe sous réserve des restrictions expliquées dans la partie suivante. Pour une BRDF anisotropique, au moins trois configurations sont nécessaires, les deux présentes dans le cas isotrope et une dernière pour la direction principale d'anisotropie, *i.e.*  $\phi_i = \frac{\pi}{2}$ .

Enfin, rien n'empêche l'utilisateur de définir des lobes pour des configurations supplémentaires. En somme, l'utilisateur peut définir un lobe pour  $\omega_r$  rasant à la surface ou encore pour des nouvelles valeurs de  $\phi_i$ . Le système se chargera de paramétrer l'interpolation des lobes pour que ces nouvelles configurations soient prises en compte.

Pour définir ces configurations additionnelles, nous prévoyons tester deux formes d'interface utilisateur :

- en laissant l'utilisateur cliquer sur un pixel de la surface de l'objet et ce pour récupérer la direction de la lumière
- en lui présentant un hémisphère 3D sur lequel les configurations déjà définies par l'utilisateur sont affichées. Un clique sur cet hémisphère autorisera l'utilisateur à définir une nouvelle configuration pour la direction de lumière sélectionnée.

Expliquons maintenant comment interpoler la forme et l'intensité du lobe entre plusieurs configurations.

### Interpolation entre les configurations

Nous commençons par expliquer l'interpolation dans un cas simplement isotropique (indépendance par rapport à  $\phi$ ) qui ne nécessite que deux configurations. Soit  $C(\phi_s, \theta_i)$  la courbe définie par l'utilisateur pour la première configuration. Dans cette première configuration  $\theta_i = 0$ . Similairement notons  $C(\phi_s, \theta_0)$  la courbe définie par l'utilisateur pour la seconde configuration ( $\theta_0$  est choisi par l'utilisateur). Un schéma d'interpolation qui transforme de manière continue  $C(\phi_s, \theta_0)$  en  $C(\phi_s, \theta_i)$  est donné par :

$$C(\phi_s, \theta) = C(\phi_s, \theta_0) + (C(\phi_s, \theta_i) - C(\phi_s, \theta_0))f(\theta) \quad (2.19)$$

où la fonction  $f(\theta)$  est donnée par :

$$f(\theta) = \begin{cases} \frac{\cos \theta - \cos \theta_0}{1 - \cos \theta_0}, & \theta \leq \theta_0 \\ 0, & \theta > \theta_0. \end{cases} \quad (2.20)$$

Cette fonction contrôle la vitesse de l'interpolation et peut être définie par l'utilisateur à condition que  $f(0) = 1$  et  $f(\theta_0) = 0$ .

Pour le cas anisotropique, trois configurations au minimum sont nécessaires :

$$\begin{cases} (\theta_i, \phi_i) = (0, 0) & (\text{configuration 1}) \\ (\theta_i, \phi_i) = \left(\frac{\pi}{2}, 0\right) & (\text{configuration 2}) \\ (\theta_i, \phi_i) = \left(\frac{\pi}{2}, \frac{\pi}{2}\right) & (\text{configuration 3}). \end{cases} \quad (2.21)$$

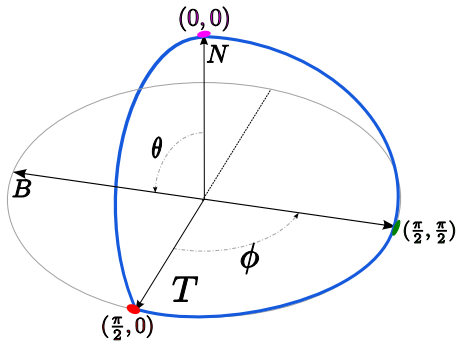


FIG. 2.16 – Exemple d’un triangle sphérique défini par 3 configurations : au sommet de la normale où  $(\theta_i, \phi_i) = (0, 0)$ , à angle rasant *i.e.*  $(\theta_i, \phi_i) = (\frac{\pi}{2}, 0)$ , et enfin dans la direction principale d’anisotropie où  $(\theta_i, \phi_i) = (\frac{\pi}{2}, \frac{\pi}{2})$ .

Nous supposons ici que la configuration définie pour  $\phi = 0$  est identique à celle pour  $\phi = \pi$  ; de même pour la configuration à  $\phi = \frac{\pi}{2}$  est identique à celle pour  $\phi = \frac{3\pi}{2}$ . Ces trois configurations forment un triangle sphérique illustré la figure 2.16. Lorsqu’on ajoute une autre configuration, elle se trouve nécessairement sur le triangle sphérique définis par les trois configurations précédentes grâce aux symétries. Tout se passe comme si ce triangle sphérique était subdivisé pour prendre en compte la nouvelle configuration.

### Implémentation

L’implémentation proposée de notre système repose sur l’utilisation du GPU et de *shaders* afin d’obtenir un système d’édition interactif. Les différents paramètres de définition de courbes et d’interpolations sont passés à un *shader* qui calcule à la volée la forme, l’énergie et la couleur du lobe. Une approche discrète où des textures 2D (pour les iso-courbes) et 1D (pour la fonction de potentiel) sont préremplies est également possible. Ceci remplacerait le coût des calculs sur GPU par des accès texture.

Afin d’éviter que l’utilisateur ne sélectionne une configuration dans la zone d’auto-ombrage de l’objet, une *shadow map* peut être calculée à la volée en fonction de la position de la lumière. Ceci est uniquement nécessaire si l’interface retenue pour ajouter des configurations supplémentaires est celle qui utilise directement l’objet.

### Intérêts et limitations

L’intérêt d’un tel paradigme est de permettre un contrôle efficace (temps réel sur GPU) et plus facile du lobe aussi bien pour ses formes que pour son gradient. La figure 2.18 montre quelques résultats préliminaires obtenus à l’aide de ce paradigme en utilisant des courbes polaires analytiques.

En revanche les formes de lobe sont limitées à la classe de fonctions des *star-shape* généralisées. Si deux couleurs différentes sont utilisées pour deux configurations, un gradient de couleur entre les deux configurations s’opèrera comme le montre la figure 2.17. Le schéma d’interpolation proposé



FIG. 2.17 – Exemple de gradient de couleur obtenu en spécifiant une couleur du lobe jaune pour la configuration  $(\theta_i, \phi_i) = (0, 0)$  et une couleur rouge pour la configuration  $(\theta_i, \phi_i) = (\frac{\pi}{2}, 0)$ .

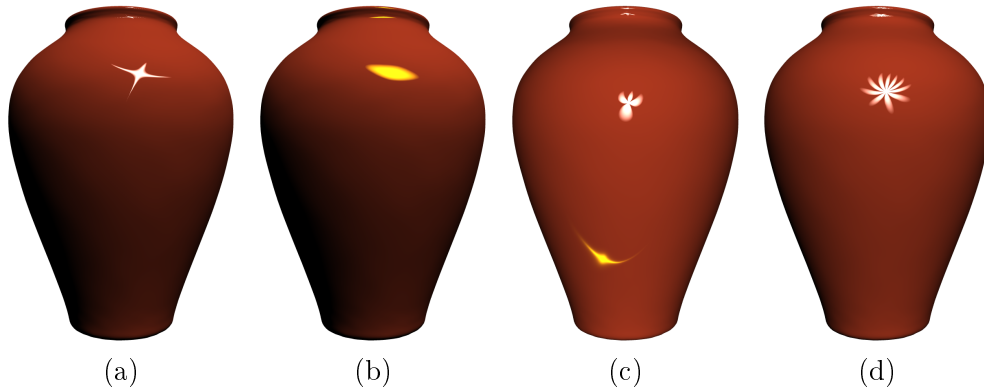


FIG. 2.18 – *Highlights* dessinés avec un courbe de type (a) *star-shape*, (b) une ellipse, (c) la fonction analytique 4-pétales et une rétro-réflexion, (d) la fonction analytique 10-pétales.

est facile à suivre, la seule difficulté provient du fait que le lobe n'est pas défini directement sur la surface et ne tient donc pas compte de la variation des normales. Le but de notre second paradigme d'édition orienté peinture est de repousser cette limitation.

## 2.2.4 Paradigme orienté image

### Édition

Dans ce paradigme l'utilisateur peint son *highlight* directement sur la surface. La spécification prend donc en compte la configuration géométrique (les normales) à la surface de l'objet.

Les outils fournis à l'utilisateur dans ce paradigme sont de type pinceaux ou brosses dont la taille et la couleur sont sélectionnables. Ce sont des outils de dessin 2D communs que l'on retrouve dans des outils tels que **GIMP**<sup>4</sup> ou **Photoshop**<sup>5</sup>. L'utilisateur peint dans le plan image mais uniquement pour les pixels qui correspondent à une surface visible d'un objet qui ne sont pas dans l'ombre.

<sup>4</sup>GIMP : <http://www.gimp.org>

<sup>5</sup>Photoshop : <http://www.adobe.com/fr/products/photoshop/>



Lorsque l'utilisateur remplit un pixel à l'aide d'une couleur, il assigne une valeur (la couleur) de BRDF pour les directions de lumière et de vue échantillonnées au centre du pixel. L'utilisateur remplit donc peu à peu une structure qui représente la BRDF et le système se charge d'extrapoler les valeurs manquantes de la BRDF pour les configurations non remplies. Si l'utilisateur modifie la direction de la source de lumière ou de vue, il verra les valeurs interpolées. Ces valeurs sont elles-mêmes éditables par l'utilisateur qui peut les changer en repeignant par-dessus.

Afin d'accélérer le processus de création de BRDFs, la structure représentant la BRDF peut être préremplie à partir d'un modèle analytique choisi par l'utilisateur. Ceci permet également d'accélérer le processus d'interpolation des valeurs.

L'utilisateur peut aussi utiliser une photographie pour appliquer un *highlight* une fois sa zone de définition spécifiée ainsi que les directions de lumière et de vue. L'utilisateur peut choisir de récupérer uniquement les valeurs d'intensité du lobe ou aussi les valeurs colorimétriques.

### Représentation et encodage de la BRDF

La représentation utilisée est une matrice 4D remplie par l'utilisateur au fur et à mesure qu'il peint sur la surface. En supposant, au départ, que la structure n'a pas été préremplie, elle passe d'une taille vide, à creuse, puis éventuellement à dense.

Les valeurs manquantes dans la structure doivent être complétées. Nous nous proposons d'adapter une technique basée sur le mécanisme de diffusion de Poisson [PGB03, Aga07] utilisée en imagerie 2D ou encore en reconstruction de surface [DHDS05]. Le mécanisme de Poisson garantit la continuité du gradient. Rendu populaire par Pérez et al. [PGB03], Le mécanisme de Poisson fusionne une image dans une autre en minimisant la visibilité des raccords ou coutures (*seamless cloning*). Poisson a aussi été utilisé pour combler dans une grille tridimensionnelle des normales d'une surface qui sont absentes [DHDS05] ou données mal acquises par un scanner 3D.

Afin de pouvoir appliquer le mécanisme de Poisson à un ensemble de données manquantes, il est nécessaire de connaître le gradient des données. Dans notre cas, une étape de diffusion du gradient [XP98, ES04] est nécessaire. Cette technique, appelée *Gradient Vector Flow* (GVF), permet de propager et de lisser le gradient initial sous l'action du Laplacien. La force de la propagation et du lissage est contrôlée par un paramètre global. Lorsqu'il est nul, aucune propagation ne se fait (le gradient initial est conservé) tandis que lorsqu'il tend vers l'infini, le gradient initial est transformé en champ uniforme dont la valeur est la moyenne du gradient initial. Notons qu'il est nécessaire de modifier (cf. [DHDS05]) la technique de GVF afin de conserver (sans les lisser) les valeurs entrées par l'utilisateur. Enfin, on peut construire ce champ et en parallèle appliquer le processus de Poisson pour garantir que la propagation sera continue aux valeurs frontières (celles entrées par l'utilisateur).

Il est primordial dans un contexte d'édition interactive que le processus de Poisson se réalise le plus rapidement possible. Ceci pourrait être fait en utilisant un solveur multigrille [Pop03] qui tourne en parallèle sur CPU, voire sur GPU [BFGS03]. A l'heure actuelle, les techniques multigrilles sont réputées être les plus rapides pour résoudre les équations aux dérivées partielles dans le domaine discret. Le principe est de discrétiser le problème (grille hiérarchique), de le résoudre à un niveau plus faible, et ensuite de repropager à un niveau plus fin.

D'autres techniques pourraient être envisagées pour remplir la matrice creuse sous réserve qu'elles garantissent :

- une continuité de la variation des valeurs sans modification des valeurs entrées par l'utilisateur
- une mise à jour et/ou un calcul rapide des valeurs manquantes.

Pour encoder notre structure 4D, nous prévoyons plusieurs possibilités :

- Dans une texture 3D pour une BRDF isotropique ou après une compression en utilisant par exemple les *Polynomial Texture Maps* (PTM) pour une BRDF anisotropique. Le principe des PTM est d’encoder dans un polynôme la variation de la luminosité contenue dans une texture plutôt que de la stocker explicitement.
- Une structure 4D mappée directement sur GPU à l’aide de **glift** [LSK<sup>+</sup>06] si les performances GPU correspondent à nos besoins.
- Deux textures 2D en utilisant la factorisation homomorphique de McCool et al. [MAA01].

### Évaluation des différentes paramétrisations

Nous prévoyons tester différentes paramétrisations, incluant celle proposée pour le paradigme d’édition vectorielle. Pour ce faire, on effectuera un test qualitatif du remplissage de la structure lorsque l’utilisateur peint sur la surface. Il s’agit de vérifier comment (de manière creuse ou dense) les différentes paramétrisations affectent le remplissage de la structure. Cette évaluation est nécessaire pour s’assurer de préconditionner la structure à l’étape d’interpolation.

### Validation de la technique d’interpolation

Nous effectuerons un test quantitatif et un test qualitatif pour valider la méthode d’interpolation des données manquantes de la structure 4D.

Le test quantitatif consiste à remplir la structure 4D à partir d’un modèle analytique connu, puis de décimer des valeurs dans la structure, d’effectuer la reconstruction à l’aide de la technique d’interpolation choisie, et de mesurer ensuite la différence entre la structure originale et la structure reconstruite. On peut effectuer différents tests suivant le nombre de valeurs décimées pour avoir un meilleur aperçu du comportement de la technique d’interpolation suivant la densité de valeurs manquantes et la paramétrisation. Cette même différence de reconstruction sera aussi mesurée sur une surface pour évaluer comment l’erreur varie en fonction de la normale. L’utilisation d’un modèle paramétrique de BRDF sert uniquement ici à mesurer l’efficacité et à départager les différentes techniques d’interpolation.

Le test qualitatif consiste à observer la qualité de la reconstruction non seulement obtenue dans la structure mais également sur la surface. Il permettra également d’observer la cohérence de la reconstruction dans une séquence animée où les directions de vue et de lumière varient.

### Problèmes envisagés

Nous énumérons ici les différentes difficultés et problèmes que ce paradigme soulève.

Si l’utilisateur veut spécifier des détails fins pour un *highlight*, il ne veut pas forcément que la technique d’interpolation le diffuse et le lisse. Il faudra étudier comment modifier le processus de remplissage pour répondre à cette difficulté.

Lorsque l’utilisateur dessine sur une surface complexe, il se peut qu’il dessine de manière différente à deux endroits différents du point de vue de la position sur la surface, mais identiques pour les configurations lumière-vue. Il y a donc une ambiguïté que seul l’utilisateur peut lever. S’il veut conserver les différences, l’utilisateur est en train de définir une SVBRDF et plusieurs solutions techniques sont alors possibles pour gérer la SVBRDF.

## Chapitre 3

# Structure pour l'éclairage indirect

Dans ce chapitre nous nous intéressons à l'édition de l'éclairage indirect qui constitue une autre facette de l'apparence. La figure 3.1(b) montre à quel point l'éclairage indirect est important dans certains endroits de la scène. Par exemple, la sphère sous la table est difficilement visible sans prise en compte de l'éclairage indirect. La première partie de ce chapitre (section 3.1) rappelle les formulations mathématiques qui définissent l'éclairage indirect. Nous passons ensuite brièvement en revue les différents types d'algorithmes d'éclairage qui permettent de le calculer. Nous mettons un accent sur les techniques de *caching* (section 3.2) qui sont très utilisées pour accélérer le calcul de l'éclairage indirect en mettant en avant leurs forces et leurs faiblesses. Nous proposons une nouvelle structure (section 3.3) servant non seulement de structure de cache mais permettant aussi l'édition de l'éclairage indirect (section 3.5). Contrairement aux approches précédentes d'édition de l'éclairage (section 3.4) nous optons pour un paradigme d'édition qui ne nécessite pas de recalculer l'ensemble de l'éclairage indirect à chaque édition.

### 3.1 Introduction à l'éclairage global en synthèse d'images

#### 3.1.1 Approche intuitive

Les algorithmes de rendu avec éclairage global prennent en compte l'ensemble des interactions possibles entre la lumière et la géométrie de la scène. Ils tiennent compte non seulement de l'éclairage direct (éclairage dit local) d'un objet par les sources lumineuses, mais aussi de l'éclairage provenant d'autres objets (éclairage indirect) de la scène réfléchissant potentiellement de la lumière. La figure 3.1 exhibe, pour une même scène, la différence entre un rendu obtenu en tenant compte de l'éclairage local vs éclairage global.

#### 3.1.2 Formulation mathématique du rendu

En première approximation, un algorithme de rendu doit calculer pour chaque pixel d'une image une valeur de radiance (cf. annexe A).

Kajiya [Kaj86] a montré que tout algorithme de rendu tentait en fait de résoudre une même équation appelée "équation du rendu", qui découle de la loi de la conservation de l'énergie. La loi de la conservation de l'énergie en un point  $x$  peut s'exprimer par :

$$L(x \rightarrow \omega_o) = L_e(x \rightarrow \omega_o) + L_r(x \rightarrow \omega_o) \quad (3.1)$$

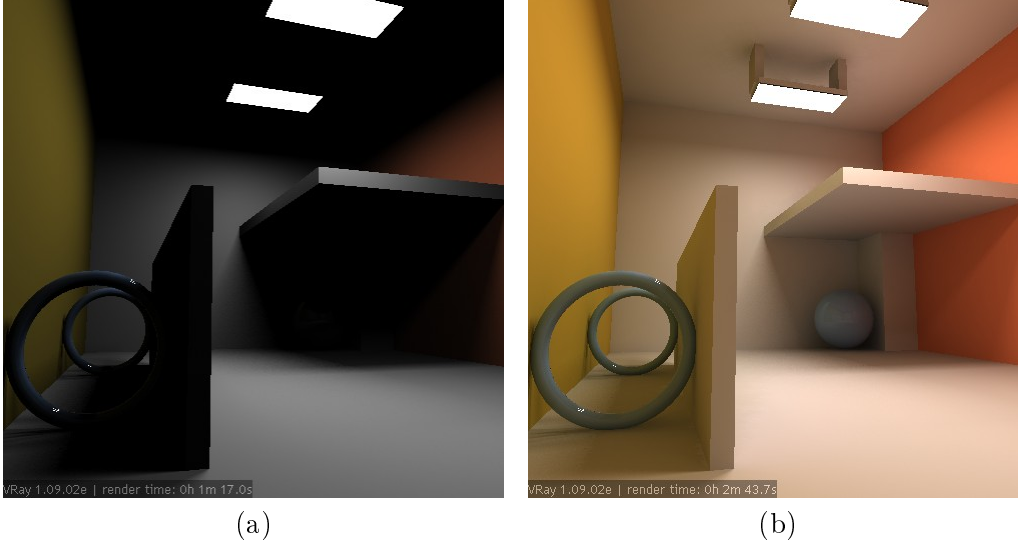


FIG. 3.1 – Différence entre éclairage local et éclairage global. (a) Eclairage local. (b) Eclairage global. Images tirées de [htt05].

où :

- $L_e(x \rightarrow \omega_o)$  la radiance émise par la surface au point  $x$  dans la direction  $\omega_o$
- $L_r(x \rightarrow \omega_o)$  la radiance réfléchiée par la surface au point  $x$  dans la direction  $\omega_o$ .

En réexprimant la radiance réfléchiée, on obtient l'équation du rendu [Kaj86] :

$$L(x \rightarrow \omega_o) = L_e(x \rightarrow \omega_o) + \int_{\Omega_N} L(x \leftarrow \omega_i) f_r(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i \quad (3.2)$$

où  $f_r(x, \omega_i, \omega_o)$  représente la BRDF au point  $x$ .

D'apparence simple, cette équation est très complexe puisque l'inconnue (la radiance) apparaît dans les deux membres de l'équation ; ceci met en évidence son caractère hautement récursif. Tout algorithme de rendu résout de manière plus ou moins partielle et plus ou moins efficace l'équation du rendu.

### 3.1.3 Éclairage indirect

Pour mettre en évidence les notions d'éclairage indirect et d'éclairage local, on décompose l'intégrale de l'équation du rendu en deux termes. Si l'on fait l'hypothèse que les sources de lumière ne font qu'émettre de la lumière, l'équation précédente devient :

$$L(x \rightarrow \omega_o) = L_e(x \rightarrow \omega_o) + \underbrace{\int_{\Omega_N \setminus \Omega_D} L(x \leftarrow \omega_i) f_r(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i}_{\text{Eclairage indirect}} + \underbrace{\int_{\Omega_D} L(x \leftarrow \omega_i) f_r(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i}_{\text{Eclairage direct}} \quad (3.3)$$

où :

- $\Omega_D$  l'ensemble des angles solides formé par les sources de lumière

–  $\Omega_N \setminus \Omega_D$  représente l'ensemble des angles solides formé par la différence entre  $\Omega_N$  et  $\Omega_D$ .

Pour montrer le caractère non récursif de l'intégrale de l'éclairage direct, on réécrit cette intégrale en faisant apparaître la radiance émise depuis les sources lumineuses. Pour ce faire, on transforme l'intégrale hémisphérique en intégrale surfacique :

$$\int_{\Omega_D} L(x \leftarrow \omega_i) f_r(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i = \int_A L(y \rightarrow -\omega_i) f_r(x, \omega_i, \omega_o) V(x, y) G(x, y) dA \quad (3.4)$$

où  $y$  est un point sur une source de lumière. La fonction de visibilité  $V(x, y)$  entre  $x$  et  $y$  est définie par  $V(x, y) = 1$  si  $x$  et  $y$  sont mutuellement visibles, 0 dans le cas contraire. Le terme  $G(x, y)$  est le facteur géométrique défini par :

$$G(x, y) = \frac{\cos(N, \omega_i) \cos(N_y, \omega_i)}{\|x - y\|^2} \quad (3.5)$$

où  $N_y$  est la normale au point  $y$ .

Dans tous les cas, l'intégrale qui calcule l'éclairage direct n'est pas récursive et pose moins de difficultés à résoudre que l'intégrale de l'éclairage indirect. C'est cette dernière à laquelle s'attaque les algorithmes d'éclairage global. Étant données la complexité des équations et la difficulté à les résoudre de manière efficace, l'édition de l'éclairage dans un contexte global est loin d'être évidente et l'édition inverse l'est encore moins. Avant d'aborder les systèmes existants qui permettent d'éditionner l'éclairage (cf. section 3.4), nous passons brièvement en revue les techniques classiques qui permettent de calculer l'éclairage global ainsi que les techniques de *caching*<sup>1</sup>. Nous verrons que les techniques de *caching* permettent d'accélérer les temps de rendu au prix de l'introduction d'un biais dans la solution calculée.

### 3.1.4 Aperçu des algorithmes d'éclairage global

Historiquement les premiers algorithmes font l'hypothèse que la scène est diffuse et résolvent de manière déterministe l'ensemble des interactions possibles entre les divers éléments de la scène. Ces algorithmes, basés sur les méthodes d'éléments finis, se regroupent sous les méthodes à base de radiosité. Un aperçu de ces méthodes se trouve dans [SP94]. Ils ont l'avantage d'être déterministe, d'être indépendant du point de vue, et leur résultat peut donc être réutilisé. Malheureusement, ils nécessitent une géométrie subdivisible et ne prennent pas en compte des scènes avec des BRDFs *glossy*, voire miroir.

A l'opposé, d'autres techniques stochastiques utilisent les méthodes d'intégration de Monte-Carlo pour calculer une solution de l'équation du rendu. Le principal avantage de ces méthodes est leur simplicité d'implémentation et leur généralité, car toutes les BRDFs sont utilisables et l'ensemble des interactions sont prises en compte. Leur principal inconvénient est leur lenteur à converger vers une solution non bruitée. Un aperçu des méthodes stochastiques est donné dans [DBB06]. Les techniques stochastiques échantillonnent les différents chemins (*path*) que peut prendre la lumière pour calculer la radiance dans un pixel. La méthode la plus simple et la plus connue, le *ray tracing* [Whi80], consiste à lancer des rayons depuis une caméra et à les propager dans la scène. Les rayons se propagent uniquement pour la réflexion spéculaire parfaite et la réfraction ; par conséquent seul un sous-ensemble des interactions lumineuses sont prises en compte. Une généralisation du *ray tracing*,

<sup>1</sup>Le terme "*caching*" désigne la technique tandis que le terme "cache" désigne la structure qui permet d'implémenter la technique.

appelée *path-tracing*, prend en compte l'ensemble des chemins lumineux ; elle a été introduite par Kajiya [Kaj86] comme solution à son équation du rendu. Une optimisation du *path-tracing*, qui couple des rayons partant de la caméra avec d'autres partant de la lumière, a été proposée par Lafortune et Willems [LW93] et Veach et Guibas [VG94, VG97]. Toutes ces méthodes sont sujettes au bruit. Le bruit provient de divers facteurs dont l'un des principaux est l'évaluation de l'intégrale de l'éclairage indirect. Ces diverses méthodes le réduisent de manière plus ou moins efficace.

Afin d'accélérer le calcul de l'éclairage indirect, des techniques de *caching* ont été développées. Le principe de ces techniques est de précalculer l'éclairage indirect à certains endroits de la scène et d'interpoler aux endroits manquants à partir des valeurs précalculées. Ceci accélère le temps de calcul et a pour effet de remplacer le bruit, visuellement déplaisant, par un biais (du flou) qui est plus acceptable pour notre système visuel.

### 3.1.5 Notion de *caching*

La stratégie des techniques de *caching* est de précalculer l'éclairage incident à différents endroits et de l'interpoler aux endroits manquants. Malheureusement, il est complexe de représenter l'éclairage incident de manière efficace quelque soit la propriété de réflectance des surfaces intervenant dans l'éclairage. Cette difficulté est liée au large éventail de BRDFs. Les besoins de représentations de l'éclairage incident ne sont pas les mêmes pour les BRDFs basses fréquences (lambertiennes) indépendantes du point de vue que pour les BRDFs spéculaires (miroir) qui sont entièrement dépendantes du point de vue. Ainsi les stratégies pour représenter l'éclairage incident sont souvent dépendantes de la BRDF sous-jacente du matériau. Aussi, les premiers systèmes de cache étaient principalement utilisés pour les BRDFs diffuses. Nous en donnons ci-dessous la justification mathématique.

Si l'on considère qu'une BRDF peut être séparée en une composante diffuse et une composante spéculaire, nous pouvons réécrire l'équation de l'éclairage indirect par :

$$\begin{aligned}
& \int_{\Omega_N} L(x \leftarrow \omega_i) f_r(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i = \\
& \int_{\Omega_N} L(x \leftarrow \omega_i) f_{r,S}(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i + \int_{\Omega_N} L(x \leftarrow \omega_i) f_{r,d}(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i = \\
& \int_{\Omega_N} L(x \leftarrow \omega_i) f_{r,S}(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i + f_{r,d}(x) \int_{\Omega_N} L(x \leftarrow \omega_i) \cos(N, \omega_i) d\omega_i = \\
& \int_{\Omega_N} L(x \leftarrow \omega_i) f_{r,S}(x, \omega_i, \omega_o) \cos(N, \omega_i) d\omega_i + f_{r,d}(x) E(x)
\end{aligned} \tag{3.6}$$

où  $f_{r,S}$  représente la composante spéculaire (pas nécessairement spéculaire parfaite) de la BRDF et  $f_{r,d}$  la composante lambertienne (donc le cas diffus uniforme). Puisque la composante diffuse est constante, on peut la sortir de l'intégrale. La quantité restante est appelée irradiance au point  $x$ , souvent notée  $E(x)$ . Dans le cas d'une BRDF diffuse, il y a donc un résultat exact qui permet en fonction de l'irradiance de calculer la radiance réfléchie :

$$L(x \rightarrow \omega_o) = f_{r,D}(x)E(x) \quad \text{avec} \quad E(x) = \int_{\Omega_N} L(x \leftarrow \omega_i) \cos(N, \omega_i) d\omega_i. \tag{3.7}$$

C'est là toute la stratégie des premiers schémas de *caching* : précalculer  $E(x)$  à différents endroits et l'interpoler aux autres parce que recalculer l'intégrale de l'irradiance pour chaque pixel est extrêmement coûteux si l'on veut éliminer le bruit.

Le *caching* est énormément utilisé dans les logiciels commerciaux pour obtenir plus rapidement un rendu en éclairage global d'une scène. La technique la plus utilisée est celle proposée par Ward et Heckbert [WH92] (ou une de ses dérivées) que nous détaillerons à la prochaine section.

Comme nous venons de le voir, l'équation du rendu, et plus particulièrement celle de l'éclairage indirect, est très coûteuse à évaluer. Il n'est pas possible, sauf technique spécialisée abordée à la section 3.4 de recalculer en temps réel l'ensemble de l'image si l'on modifie les sources de lumière. Dans le contexte d'édition d'éclairage, nous devons donc développer une structure capable d'approximer rapidement la fonction d'éclairage (contrainte de rendu en temps interactif) mais pouvant aussi être modifiée de manière simple et intuitive (contrainte d'édition).

À la section 3.3, nous proposons une nouvelle structure pour stocker et éditer l'éclairage indirect. À la différence des techniques de *caching* antérieures abordées à la prochaine section, elle possède de bonnes propriétés qui permettent une édition de l'éclairage indirect.

## 3.2 Problématique et travaux antérieurs en *caching*

Les points importants d'un algorithme de *caching* sont :

- sa robustesse à reconstruire le type d'éclairage incident ainsi que son indépendance par rapport à la BRDF. Ceci est directement lié à la quantité radiométrique stockée et à la représentation utilisée.
- la robustesse lorsqu'on augmente les détails géométriques.
- la continuité du schéma d'interpolation qui lui est associé.
- la consommation mémoire et l'adaptabilité vis-à-vis de la taille de la scène utilisée.

### *Photon Mapping*

Le *photon map* introduit par Jensen [Jen01] est une structure de cache qui stocke des particules (photons). Les photons sont émis à partir des sources de lumière et stockés lors d'une première passe préalable au rendu de la scène. Un *kd-tree* stocke en 3D l'impact des photons sur la géométrie de la scène. C'est donc une structure mais qui "suit" la géométrie de la scène. Lorsque l'éclairage indirect doit être évalué à une position donnée, une recherche de photons dans un voisinage, paramétré par l'utilisateur, est effectuée. La densité de photons permet alors de calculer l'éclairage indirect. Comme les photons ne sont pas organisés, une reconstruction de mauvaise qualité est obtenue lorsqu'on accède directement au *photon map*. Pour remédier à ce problème la structure de cache est interrogée de manière indirecte lors du rendu final (*final gathering*). La figure 3.2 montre la différence de qualité obtenue entre un accès direct au *photon map* et avec une utilisation du *final gathering*.

On obtient au prix d'un coût élevé de temps de rendu une très bonne qualité de reconstruction de l'éclairage indirect. En général, on utilise le mécanisme de Christensen [Chr99] qui précalcule l'irradiance sur les photons ou celui de Ward (cf. ci-dessous) pour accélérer le rendu.

### *Caching* de Ward et al.

Le *caching* de Ward et al. [WRC88] stocke l'irradiance en certains points de la scène. Dans une première étape, la position et le nombre d'échantillons sont déterminés à l'aide d'une fonction

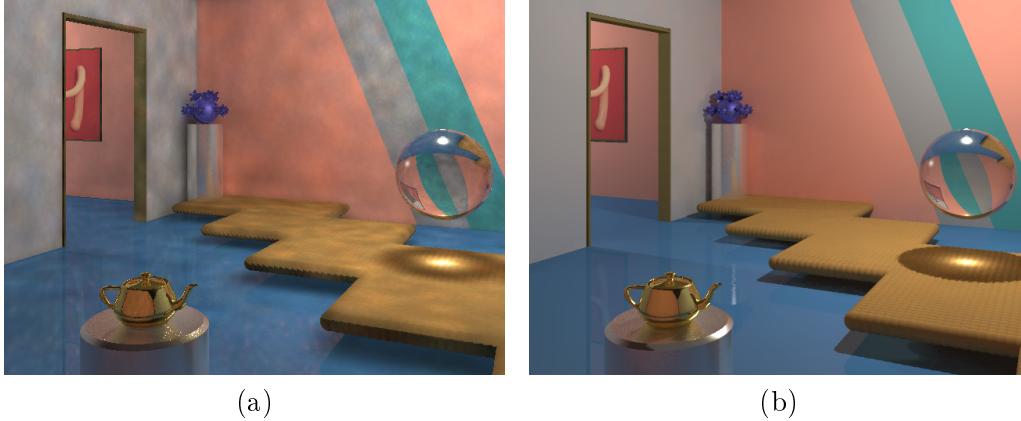


FIG. 3.2 – Rendus avec *photon mapping*. (a) Utilisation directe du *photon map*. (b) Utilisation du *final gathering*. Images tirées de [Jen96].

dépendante de critères géométriques (variation des normales et proximité d'autres objets) et du point de vue de la scène. Dans une seconde passe, l'irradiance est interpolée en un point en prenant les échantillons voisins dans le cache qui lui sont proches. La valeur stockée par échantillon est l'irradiance, grandeur dépendante de la normale et qui doit donc être recalculée dès que la normale varie.

Le coefficient d'albédo diffus peut donc être modifié dynamiquement, comme le montre l'équation 3.7, permettant ainsi une certaine édition. La continuité de la reconstruction, du schéma original de Ward n'est pas garantie parce que les échantillons ne sont pas organisés de manière spatiale. L'utilisation de gradients rotationnels et translationnels [WH92] améliore la qualité de la reconstruction mais des discontinuités peuvent encore survenir comme le montrent les images de la figure 3.3.

La fonction de positionnement des échantillons dépend de la normale, ce qui pose des problèmes sur la concentration des échantillons dès que de nombreux détails géométriques ou concavités sont présents dans la scène (cf. figure 3.4(c)).

### Extension de Tabellion et Lamorlette au *caching* de Ward

Le mécanisme de *caching* proposé par Tabellion et Lamorlette [TL04] vise à améliorer le positionnement des échantillons, et notamment à éviter une accumulation dans les coins en modifiant la fonction de positionnement des échantillons, comme le montrent les figures 3.4(a) et (b). De plus, lors de la première étape, ils n'utilisent pas la géométrie détaillée (*bump maps* ou *displacement maps*)<sup>2</sup> mais une version simplifiée (maillage basse définition) pour limiter le nombre d'échantillons. Cependant, la fonction de positionnement des échantillons reste dépendante des critères géométriques et ne limiterait donc pas le nombre élevé d'échantillons sur un maillage haute définition.

Bien que cela ne soit pas totalement clair, il semblerait que Tabellion et Lamorlette stockent non seulement une valeur d'irradiance mais également une direction de lumière incidente moyenne afin de pouvoir utiliser n'importe quelle BRDF (bien que cela soit physiquement incorrect pour certaines BRDFs non-diffuses). La continuité du schéma de reconstruction semble grandement améliorée. Cependant, il est difficile de prouver que le schéma d'interpolation garantit une continuité supérieure à  $C^0$ .

<sup>2</sup>Le *bump map* et le *displacement map* sont des techniques qui simulent des détails géométriques à l'aide d'une texture 2D ou sous forme d'une fonction analytique.



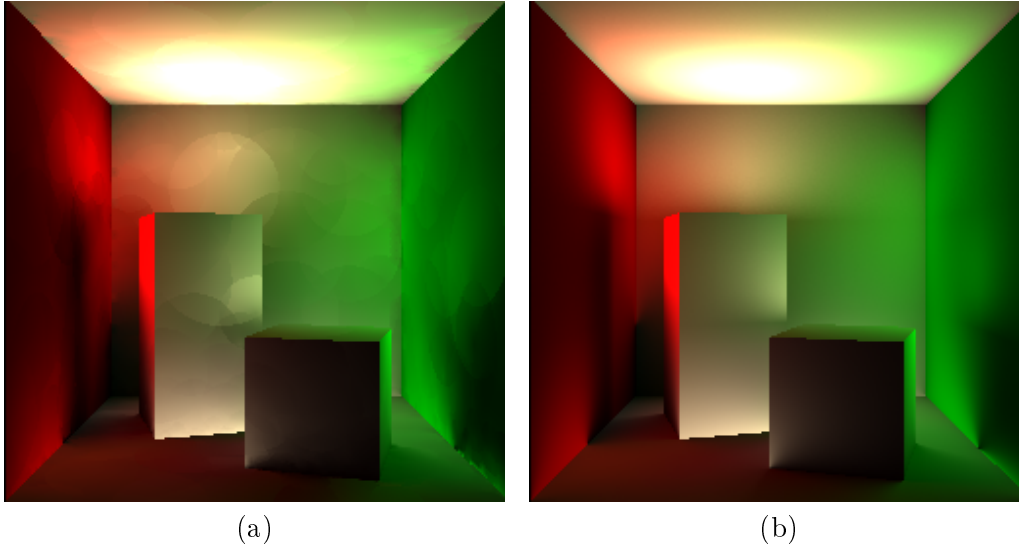


FIG. 3.3 – Comparaison entre de l'éclairage indirect reconstruit avec : (a) gradients de Ward (b) Approximation de meilleure qualité obtenue en échantillonnant l'hémisphère. Images tirées de [KGPB05].

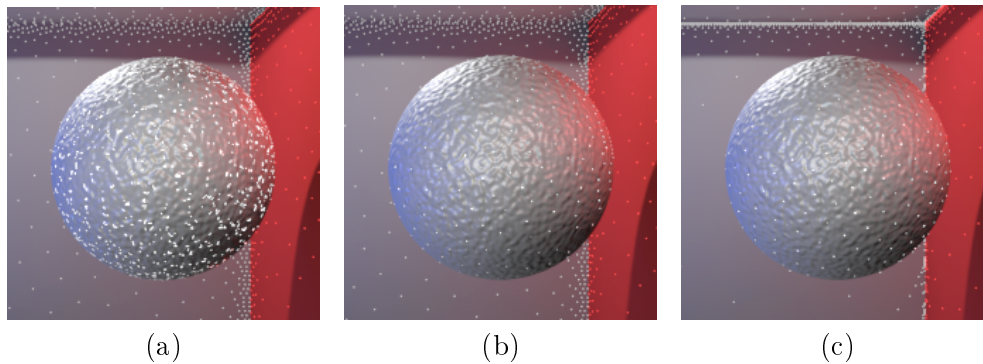


FIG. 3.4 – Comparaison du positionnement des échantillons dans le cache en fonction du critère utilisé. (a) Critère de Tabellion et Lamorlette en utilisant la géométrie détaillée. (b) Critère de Tabellion et Lamorlette en utilisant la géométrie faiblement détaillée. (c) Critère de Ward. Images tirées de [TL04].

Enfin, précisons que les techniques basées sur le critère de Ward présentent l'avantage d'être consistantes, *i.e.* l'augmentation du nombre d'échantillons fait tendre la reconstruction vers la solution exacte. Si chaque pixel possède un échantillon stockant l'irradiance, alors aucune interpolation n'est effectuée et ceci revient à calculer la solution exacte de l'irradiance.

### Caching stockant la radiance incidente

Comme mentionné ci-dessus, stocker l'irradiance n'est physiquement correct que si la BRDF est diffuse. Plus récemment, des algorithmes stockant la radiance ont été développés pour permettre d'utiliser des BRDFs plus *glossy*. Comme la radiance est uniquement valable pour un angle solide donné il faut une représentation adéquate des directions sur la sphère.

Arikan et al. [AFO05] projettent la radiance dans la base des harmoniques sphériques. Leurs

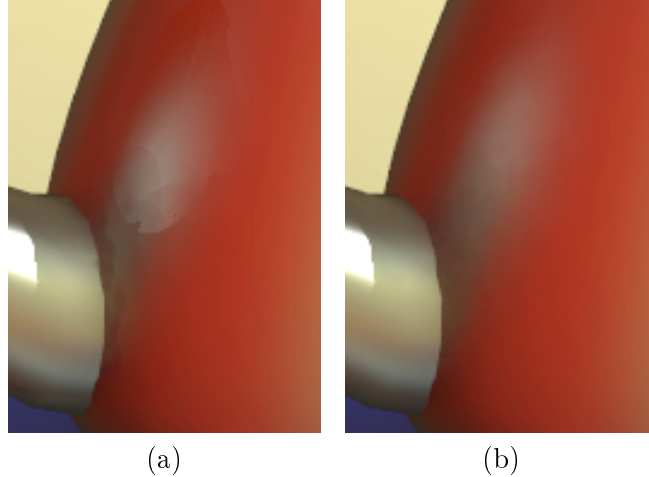


FIG. 3.5 – Comparaison de la qualité visuelle obtenue entre : (a) *caching* de Ward et al. [WH92]. (b) *caching* de Krivanek [KBPv06]. Images tirées de [KBPv06].

échantillons sont disposés dans un volume autour de la surface. L'interpolation lors de la reconstruction n'est pas garantie (mêmes raisons que pour le schéma de Ward) et surtout le critère de convergence n'est pas facilement contrôlable, voire pire, il augmente les discontinuités dans l'image finale.

Le schéma de Krivanek et al. [KGPB05] utilise également les harmoniques sphériques ou hémisphériques. Dans le cas de BRDFs *glossy*, Krivanek et al. [KBPv06] redistribuent la position des échantillons pour mieux distribuer l'erreur visuelle. Ceci a pour effet d'améliorer la qualité visuelle de la reconstruction comme le montrent les images de la figure 3.5.

Cependant, les schémas d'interpolation d'Arikan et al. et de Krivanek et al. ne positionnent pas leurs échantillons de manière régulière et la continuité de la reconstruction n'est pas garantie. Malheureusement, aussi bien la technique d'Arikan et al. que celle de Krivanek et al. ne permettent pas d'être utilisées dès que la scène contient trop de détails géométriques. De plus, comme toutes les techniques de *caching*, elles ne semblent pas être facilement exploitables pour pouvoir servir de base pour l'édition de l'éclairage indirect.

Préalablement, Greger et al. [GSHG98] ont proposé de stocker les échantillons dans un volume sur une grille régulière appelé *irradiance volume*. Contrairement au nom de la structure, c'est bien la radiance incidente qui est stockée. Comme les échantillons sont stockés sur une grille régulière suivant le type de filtre utilisé, la reconstruction peut être  $C^0$ ,  $C^1$  ou encore  $C^2$ . Greger et al. discrétisent la sphère sur 578 directions et stockent pour chacune une valeur de radiance.

### 3.3 Recherche proposée en *caching*

#### 3.3.1 Aperçu général de la solution proposée

A la lecture de ce qui a été fait précédemment en *caching*, nous pensons qu'il faut un système de cache qui :

- garantisse une continuité au moins  $C^1$  dans la reconstruction de l'éclairage
- utilise une quantité radiométrique qui permette une bonne mise à l'échelle lorsque les détails géométriques dans une scène augmentent drastiquement

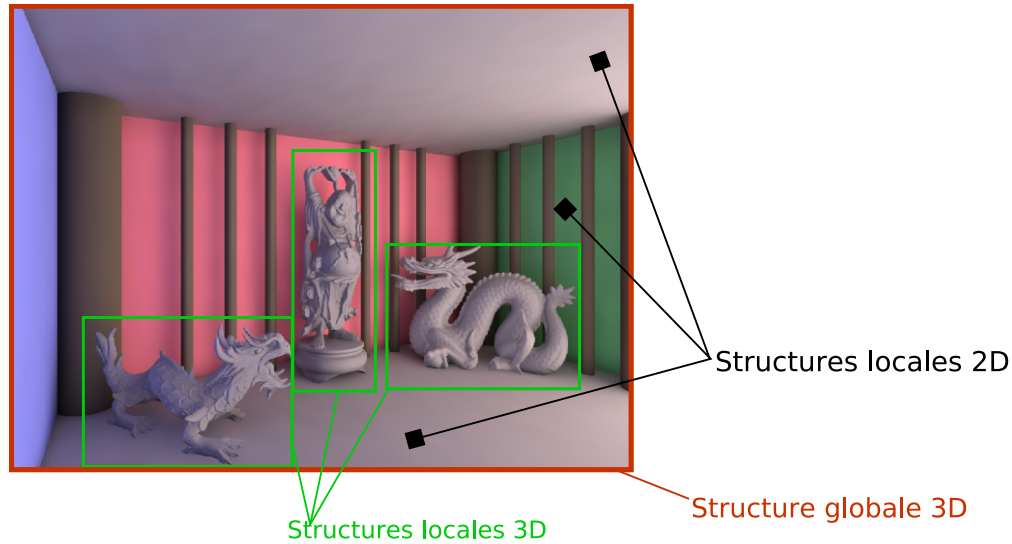


FIG. 3.6 – Illustration du type de structures utilisées en fonction du type d’objet d’une scène. En plus de la structure globale (en rouge) de la scène, des structures locales 3D (en vert) sont utilisés pour les objets volumineux tandis que des structures locales 2D sont utilisées pour les objets de types planaires.

- utilise une base de reconstruction/projection qui autorise une certaine indépendance par rapport à la BRDF utilisée (dans l’idéal du cas diffus au cas *glossy*).

Le système de caching que nous proposons est basé sur l’utilisation d’une grille régulière qui encode l’éclairage incident. Plus précisément, l’éclairage incident indirect est précalculé lors d’une première étape et stocké sur chaque sommet de la grille. Lors de l’étape de rendu, la grille est interrogée pour chaque point de la scène dont on cherche à connaître l’éclairage indirect. L’éclairage indirect pourra alors être reconstruit par interpolation. Comme illustré à la figure 3.6, nous utiliserons plusieurs types de grilles en fonction du type d’éléments contenus dans la scène afin d’améliorer la qualité de reconstruction de l’éclairage. Ceci nous permettra également une plus grande souplesse dans le cadre de l’édition. Ainsi, des grilles 2D sont utilisées pour les objets de type planaire ou quasi-planaire (plan et *bump map* ou *displacement map*) tandis que des grilles 3D sont utilisés pour les objets pleins (e.g. statues) ou semi-pleins (e.g. arbres). Ces grilles sont définies de manière locale *i.e.* dans le référentiel de l’objet auquel elles sont rattachées et elles sont immergées dans une grille 3D globale à toute la scène. Comme nous le verrons à la section 3.5 la grille globale nous permettra de transférer des modifications effectuées sur une structure locale à une autre. Enfin, nos grilles peuvent également être raffinées de manière adaptative à l’aide d’un critère ne dépendant pas de la géométrie afin d’améliorer la reconstruction et permettre un gain en mémoire.

En résumé, le système de *caching* que nous proposons possède les caractéristiques suivantes :

1. Une grille 2D ou 3D régulière ou semi-adaptative<sup>3</sup> pour pouvoir facilement effectuer une interpolation spatiale qui sera au moins  $C^1$ .
2. Les sommets stockent une grandeur radiométrique vectorielle (vecteur d’irradiance) robuste aux variations de normales et intuitive à éditer de manière directe.

<sup>3</sup>On entend ici par structure semi-adaptative une structure adaptative dont les cellules voisines n’ont pas une différence de profondeur de subdivision strictement supérieure à 1.

3. Les vecteurs d'irradiance sont indépendants de la BRDF sous-jacente.
4. Une base de vecteurs d'irradiance stockée aux sommets de la grille qui permet de reconstruire l'irradiance pour n'importe quelle normale sous-jacente.
5. Une interpolation spatiale et une interpolation directionnelle sont effectuées pour calculer l'éclairage en un point.

### 3.3.2 Construction de la structure

#### Vecteur d'irradiance (*irradiance vector*)

Pour chaque sommet de la grille (2D ou 3D), nous stockons des vecteurs d'irradiance. Le vecteur d'irradiance [Arv94] est une quantité radiométrique vectorielle définie telle que :

$$I_N(x) = \int_{\Omega_N} L(x \leftarrow \omega_i) \omega_i d\omega_i. \quad (3.8)$$

En toute rigueur, on devrait calculer un vecteur d'irradiance par longueur d'onde mais, comme pour les BRDFs ou la radiance, en général seules trois longueurs d'onde sont utilisées. Il faut donc neuf flottants pour encoder trois vecteurs d'irradiance. Notons que l'on peut facilement stocker ces trois vecteurs d'irradiance sous la forme d'une quantité colorimétrique et de trois directions. Notons  $I_{N,i}(x)$  le vecteur d'irradiance pour la  $i$ -ème longueur d'onde. On peut définir la quantité colorimétrique  $\mathcal{C}(I_N)$  des trois vecteurs d'irradiance par :

$$\mathcal{C}(I_N) = [\|I_{N,1}(x)\|, \|I_{N,2}(x)\|, \|I_{N,3}(x)\|]. \quad (3.9)$$

De même, notons  $\mathcal{D}_i(I_N)$  la  $i$ -ème direction du vecteur d'irradiance définie par :

$$\mathcal{D}_i(I_N) = \frac{I_{N,i}(x)}{\|I_{N,i}(x)\|}. \quad (3.10)$$

Notons que les directions peuvent être compressées lorsque leur écart est trop faible par exemple, en ne conservant qu'une seule direction moyenne. De plus, les directions  $\mathcal{D}_i(I_N)$  peuvent être quantifiées [Jen01] de même pour la quantité colorimétrique en utilisant le format RGBE [War91].

L'irradiance et le vecteur d'irradiance sont reliés par :

$$E(x) = N \cdot I_N(x). \quad (3.11)$$

Comme le montre l'équation précédente, le vecteur d'irradiance permet plus de souplesse par rapport à la géométrie sous-jacente lorsqu'il s'agit de calculer l'irradiance.

#### Base de vecteurs d'irradiance

Nos échantillons étant positionnés dans l'espace, ils ne sont reliés à aucune normale de surface. Autrement dit, la normale de la géométrie sous-jacente peut prendre n'importe quelle direction repérée sur la sphère unité. Or comme le montre l'équation 3.8, le vecteur d'irradiance est valide pour une certaine normale donnée. Pour pouvoir reconstruire sur l'ensemble de la sphère unité nous utilisons plusieurs vecteurs d'irradiance, soit un vecteur d'irradiance pour les six directions principales ( $\pm X$ ,  $\pm Y$ ,  $\pm Z$ ) du repère cartésien. Ces vecteurs seront notés  $I_{\pm X}$  (resp.  $I_{\pm Y}$  et  $I_{\pm Z}$ ). Intuitivement nous disposons d'un vecteur d'irradiance pour six hémisphères qui se recouvrent partiellement. On

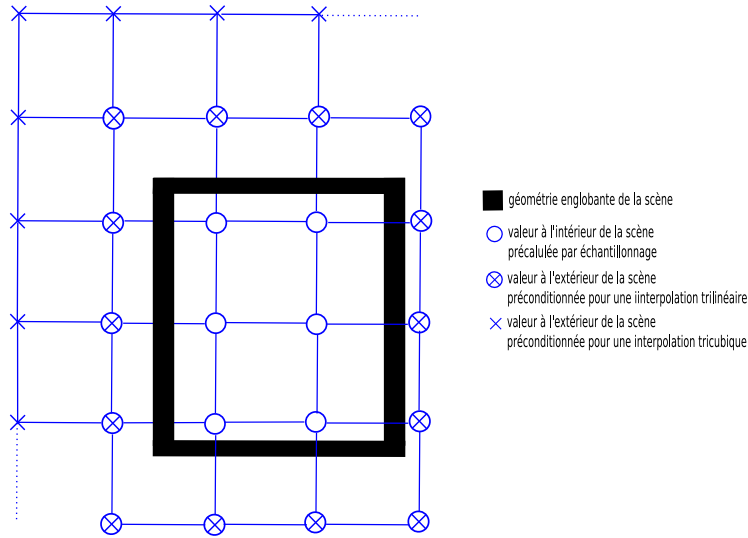


FIG. 3.7 – Illustration 2D du préconditionnement des valeurs à la frontière de la structure. La boîte englobante de la scène est représentée en traits épais noir. Les valeurs à l’intérieur de la boîte sont échantillonnées à l’aide d’un algorithme d’éclairage global. Les valeurs à l’extérieur sont par défaut nulles mais une valeur initiale leur est attribuée (préconditionnement) pour garantir une interpolation continue.

verra (cf. section 3.3.3) que pour n’importe quelle normale, seulement trois des six vecteurs d’irradiance seront utilisés. Pour calculer un vecteur d’irradiance, n’importe quel algorithme d’éclairage global (*path-tracing*, ou *bidirectional path-tracing*) peut être utilisé. On envisage éventuellement d’utiliser le GPU pour accélérer la construction de notre structure en échantillonnant pour une même direction plusieurs positions à la fois, comme déjà proposé [Hac05].

### Structures 2D et 3D

Notre structure 2D est une grille régulière ou un *quadtree* positionné légèrement en avant de la surface le long de sa normale. Cette surface peut résulter d’un *bump map*, d’une *normal map* ou d’un *displacement map*. Dans le cas d’un *displacement map* analytique, le positionnement de notre structure peut être calculé de manière exacte en s’assurant qu’elle se trouve toujours en avant des points déplacés. De plus, pour éviter les problèmes d’échantillonnage de l’hémisphère liés à un positionnement dans un coin, les échantillons sont légèrement déplacés vers le centre de leur cellule (cas adaptatif) ou de la grille (cas régulier).

Dans le cas de la structure 3D, un certain nombre d’échantillons seront positionnés sur des sommets de la grille se trouvant à l’intérieur de l’objet qui peut être opaque. Un test est effectué pour déterminer si un sommet se trouve à l’intérieur ou à l’extérieur de l’objet afin d’éviter d’échantillonner un vecteur d’irradiance dont la valeur sera nécessairement nulle. Pour ce faire, un plan médian de la géométrie contenue dans la cellule est calculé. Les échantillons positionnés à l’intérieur ne sont pas précalculés par échantillonnage mais par préconditionnement (cf. prochain paragraphe). Le même procédé est utilisé pour éviter de calculer des échantillons sur notre structure 2D qui serait recouvert par un objet.

### Préconditionnement des structures

Les échantillons se trouvant à l'intérieur (cf. figure 3.8) d'un objet n'ont pas été précalculés et ils ont par conséquent une valeur nulle. Afin d'éviter que ces valeurs n'influencent l'interpolation en forçant une sous-évaluation du vecteur d'irradiance, nous utilisons un mécanisme de Poisson pour propager à partir des valeurs précalculées des valeurs cohérentes (Poisson garantit  $C^2$ ) aux endroits où le précalcul n'était pas possible.

Un autre type de preconditionnement, appelé externe, est nécessaire pour la géométrie de la scène délimitant sa frontière externe (cf. figure 3.7). En effet, la structure 3D qui englobe toute la scène doit être légèrement plus grande que cette dernière afin de s'assurer que pour tout point de la scène une reconstruction des vecteurs d'irradiance sera possible. Cependant, les sommets de la structure qui délimitent son enveloppe ont une valeur nulle (impossible à échantillonner puisqu'ils sont hors de la scène) et vont donc causer une sous-évaluation du vecteur d'irradiance. Pour remédier à cela, nous extrapolons (par symétrie par exemple) à partir de valeurs connues, la valeur de ces sommets externes.

### Critères pour une construction adaptative

Un vaste choix peut être fait pour subdiviser notre structure. On peut utiliser un critère basé sur le type de BRDF sur l'objet, des critères géométriques, des critères de variations d'éclairage ou encore des combinaisons de différents critères. Nous choisissons de ne pas utiliser de critères géométriques afin d'éviter une explosion du nombre d'échantillons dans les coins ou sur des surfaces hautement détaillées. Nous préférons un critère reposant sur la variation de l'éclairage tel que capturée par les vecteurs d'irradiance ou encore sur le type de BRDF. Nous prévoyons utiliser le critère simple suivant : évaluation de la variance des vecteurs d'irradiance dans une même cellule, si elle est supérieure à un seuil défini par l'utilisateur, alors la cellule est subdivisée. Si la profondeur a atteint un seuil maximal dépendant de la mémoire disponible, alors la subdivision s'arrête. De plus, nous contraignons nos structures adaptatives à au plus un niveau de différence entre deux cellules voisines afin de pouvoir garantir une continuité  $C^1$  lors de la reconstruction.

### 3.3.3 Reconstruction de l'éclairage indirect

Pour un point  $x$  avec de sa normale  $N$ , nous devons effectuer une interpolation spatiale (dépendante de  $x$ ) des vecteurs d'irradiance ainsi qu'une interpolation directionnelle (dépendante de  $N$ ).

On utilise une reconstruction trilinéaire ou tricubique pour interpoler spatialement les vecteurs d'irradiance en  $x$ . Cette reconstruction est facilitée par l'utilisation de structures régulières ou semi-adaptatives. Ensuite une interpolation directionnelle sur la sphère unité est utilisée pour reconstruire le vecteur d'irradiance pour la normale  $N$ . L'interpolation directionnelle fait intervenir trois des six vecteurs d'irradiance et s'exprime par :

$$I_N(x) = I_{\pm X}(x) n_x^2 + I_{\pm Y}(y) n_y^2 + I_{\pm Z}(z) n_z^2 \quad (3.12)$$

où  $N = (n_x, n_y, n_z)$  est la normale au point  $x$ . Le choix du signe de  $X$  (resp. du signe de  $Y$  et de  $Z$ .) se fait en fonction du signe de  $n_x$  (resp.  $n_y$  et  $n_z$ ).

Dans le cas d'une BRDF diffuse, nous reconstruisons l'irradiance à l'aide du vecteur d'irradiance ainsi interpolé. Dans le cas d'une BRDF *glossy*, nous interprétons le vecteur d'irradiance comme une grandeur directionnelle. A savoir, nous considérons qu'un point est entouré de sources de lumière directionnelle locales.

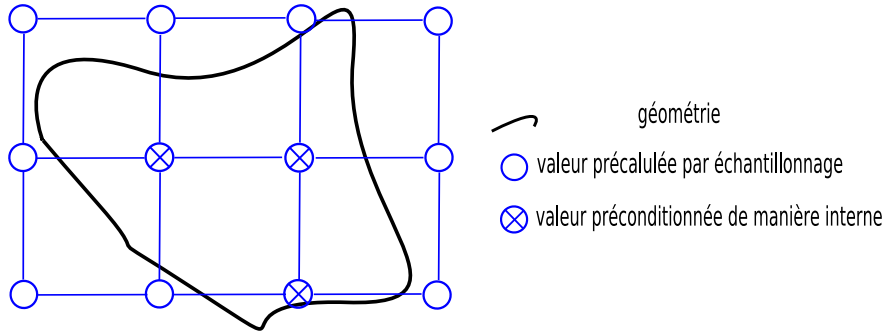


FIG. 3.8 – Préconditionnement des valeurs internes à un objet opaque.

### 3.3.4 Analyse de l'erreur

Il convient d'analyser les différentes erreurs pour mesurer l'efficacité de notre structure de cache ainsi que ses limites.

L'erreur liée au calcul d'un vecteur d'irradiance dans l'équation 3.8 décroît en  $\frac{1}{\sqrt{r}}$  où  $r$  est le nombre de rayons utilisés pour échantillonner l'hémisphère. Cette erreur est liée à l'utilisation de la technique de Monte-Carlo et elle est identique quelque soit la technique de *caching* employée.

Comme nous l'avons expliqué précédemment, nous utilisons en première approche six vecteurs (donc six hémisphères) pour représenter l'ensemble des directions possibles pour une normale. Les résultats préliminaires semblent montrer une bonne qualité de reconstruction à l'aide de six vecteurs. Nous envisageons d'étudier l'erreur de reconstruction introduite en utilisant six vecteurs. Pour ce faire, on comparera l'irradiance obtenue par interpolation directionnelle avec celle obtenue par échantillonnage pour différentes orientations de normale.

Comme tout mécanisme de cache, l'interpolation engendre un biais dans la reconstruction. L'erreur liée uniquement à l'interpolation peut se mesurer en comparant l'irradiance qui aurait été obtenue en calculant le vecteur d'irradiance au point pour lequel on a interpolé spatialement le vecteur d'irradiance.

Des mesures quantitatives sur l'ensemble du rendu sont également envisagées. À savoir, une comparaison de l'irradiance reconstruite pour l'ensemble de l'image comparée à celle obtenue par *path-tracing*. De même pour la radiance réfléchie, quantité encore plus importante puisque c'est elle qui est directement perçue par l'utilisateur.

### 3.3.5 Comparaisons avec les techniques antérieures

Nous comparerons la continuité de la reconstruction avec la technique proposée par Tabellion et Lamorlette [TL04] qui semble être celle qui produise les meilleurs résultats pour les BRDFs diffuses. Pour les BRDFs *glossy*, il serait intéressant de comparer avec la technique proposée par Krivanek et al. [KBPZ06] qui semble garantir une bonne continuité visuelle dans la reconstruction.

Nous envisageons de tester qualitativement et quantitativement l'utilisation, au sommet de la grille, d'harmoniques sphériques à la place de vecteurs d'irradiance. Afin que la comparaison soit juste, on utilisera le même nombre de coefficients pour les deux représentations. Cette comparaison nous semble importante puisque les harmoniques sphériques ont été énormément utilisées [NPG05, AFO05, KGPB05, KBPv06] pour représenter l'éclairage. D'un point de vue pratique, l'utilisation des harmoniques sphériques est plus pénalisant parce qu'il est nécessaire de projeter l'ensemble des

BRDFs de la scène dans la base des harmoniques sphériques. De plus, des rotations sont nécessaires pour aligner la représentation locale de la BRDF avec les directions de lumière qui sont décrites dans un repère global. Ces rotations constituent une pénalité en termes de performances (cf. [KKB<sup>+</sup>05]). Il nous semble aussi que la projection de l'éclairage incident dans la base des harmoniques sphériques est nettement plus coûteuse que de projeter dans notre base de vecteurs d'irradiance.

Comparativement à l'*irradiance volume* de Greger et al., notre structure peut générer des fuites alors que la représentation de Greger et al. peut créer des trous puisqu'ils discrétisent l'hémisphère sans le préfiltrer.

Enfin, Zaninetti et al. [ZSP98] utilisent les vecteurs d'irradiance non pas pour représenter l'éclairage incident mais l'éclairage réfléchi. Cette représentation est donc entièrement dépendante de la BRDF et ne permet pas autant de souplesse dans un contexte d'édition.

### 3.4 Travaux antérieurs en édition d'éclairage

Similairement à la classification que nous avons faite sur les systèmes d'édition de BRDF, les systèmes d'édition d'éclairage peuvent aussi se décomposer en plusieurs catégories :

- les systèmes interactifs d'éclairage global
- les systèmes de ré-éclairage (*relighting*) qui peuvent être considérés comme des cas particuliers des précédents
- les systèmes de *painting* (rendu inverse).

Les systèmes interactifs d'éclairage global proposent d'éditer de manière directe certains paramètres (position des lumières, position des objets) et de mettre à jour le plus rapidement possible le rendu en appliquant un algorithme d'éclairage global. Ces systèmes ont des temps de rendu qui varient de quelques secondes par image à 30 images par seconde (*frames per second*, fps). Les temps de précalcul sont eux aussi variables suivant la stratégie adoptée. Par exemple dans [GD01], une solution hiérarchique de radiosité est calculée (une seule fois) et les phénomènes lumineux de hautes fréquences tels les caustiques sont mis à jour à la volée. Le système proposé par Kristensen et al. [KAMJ05] nécessite un précalcul substantiel pour pouvoir déplacer ensuite en temps réel les sources de lumière dans des scènes quelconques. Dans le *Shading Cache* [TPWG02], une structure hiérarchique, gérée en espace objets, permet de déplacer objets et lumières en temps interactif sans aucun précalcul. L'éclairage indirect ainsi que les ombres sont mis à jour de manière incrémentale au bout de quelques secondes. Historiquement, les systèmes de *relighting* fixent la direction de vue afin d'accélérer la mise à jour de la solution. L'édition permise à l'utilisateur est de déplacer les sources de lumière. Les premiers systèmes tels que [GH00, PVL<sup>+</sup>05] ne prenaient pas du tout en compte l'éclairage indirect. Celui d'Havsan et al. [HPB06] remédie à ce problème en calculant une matrice de transfert de l'éclairage direct en éclairage indirect. Le système le plus abouti qui permet de déplacer, à maximum 7 fps, les sources de lumières, les objets et modifier les BRDFs en tenant compte de l'éclairage indirect est celui introduit par Sun et al. [SZC<sup>+</sup>07]. Dans ce système, l'édition d'un modèle de BRDF analytique est également possible.

L'intérêt de ces systèmes réside dans une prévisualisation rapide du rendu d'une scène qui serait au plus proche de ce qu'une simulation physique donnerait après de longues heures de calcul, tout en ayant la possibilité d'ajuster des paramètres. Un exemple typique d'utilisation, dans le domaine de l'architecture, est le placement de sources de lumière dans une pièce qui n'a pas encore été construite. Malheureusement, étant donné que l'éclairage global (et donc indirect) est un phénomène global, il reste difficile de spécifier localement une forme d'éclairage en ne faisant qu'ajuster des paramètres.



Le processus d'ajustement des paramètres est pénible pour l'utilisateur qui effectue une série d'essais jusqu'à être satisfait du résultat. L'intérêt des systèmes de rendu inverse est de faciliter la tâche de l'utilisateur en lui laissant spécifier directement son résultat.

Le plus souvent, dans un contexte de rendu inverse, l'utilisateur peint le résultat qu'il souhaite obtenir directement sur la surface d'un objet. Le système se charge alors d'effectuer une résolution inverse. Cette résolution inverse est en général trop complexe à résoudre (problème sous-dimensionné) pour porter sur l'ensemble des paramètres de la scène. Ainsi dans le système de Schoeneman et al. [SDS<sup>+</sup>93] les seuls paramètres que le système inverse retrouve (minimisation des moindres carrés) sont : les intensités des sources de lumière et leurs couleurs. De cette manière, le problème inverse à résoudre devient linéaire. Cependant, il n'y a pas toujours de solutions aux contraintes dessinées par l'utilisateur. De plus, lorsque l'utilisateur spécifie qu'il souhaite modifier une zone de la scène, d'autres zones vont également voir leur luminosité changer par effet de bord. Ceci est dû au fait que l'éclairage est un phénomène global. Une solution serait alors d'ajouter des contraintes supplémentaires. Malheureusement, plus on ajoute de contraintes, plus l'espace de solutions possibles se restreint.

Le système de Kawai et al. [KPC93] permet à un utilisateur de spécifier plus de contraintes, mais la métaphore de spécification est nettement moins précise. A savoir, l'utilisateur ne peint pas sur la surface mais définit des objectifs globaux de perception lumineuse tels que "je voudrais une ambiance plus feutrée". Une extension du travail de Kawai et al. [KPC93] a été proposée par Costa et al. [CSF99] où le système tente de retrouver aussi la géométrie des sources de lumière et autorise l'utilisateur à spécifier des contraintes complexes. Plus récemment, Pellacini et al. [PBMF07] ont proposé un système où la position des sources de lumière n'est pas fixée (processus inverse non-linéaire). En se restreignant à un modèle d'éclairage local, le système est alors capable de spécifier de manière interactive les différents paramètres, au sens cinématographique [Bar97], des sources de lumière. Comme l'optimisation linéaire est complexe à résoudre, Pellacini et al. [PBMF07] laissent l'utilisateur ajouter des lumières (une par une) au fur et à mesure plutôt que d'essayer de résoudre les contraintes sur un ensemble de sources de lumière prédéfinies. L'utilisateur a donc des outils d'édition directe (ajout d'une nouvelle lumière) et des outils d'édition inverse (le *painting* sur les surfaces).

En somme, il y a peu de travaux réalisés sur le rendu inverse comparé au rendu direct parce que le problème est mathématiquement difficile à résoudre. Dès lors, il est important de subdiviser les outils d'édition proposés à l'artiste pour mieux conditionner la résolution inverse. Une autre composante importante de l'édition inverse est comment réussir à convaincre l'utilisateur que les "pinceaux" qui lui sont proposés correspondent à sa façon de spécifier son résultat. De plus, il semble qu'aucun système de rendu inverse n'ait attaqué de front l'édition de l'éclairage indirect en temps interactif.

Nous proposons deux paradigmes d'édition de l'éclairage indirect qui utilisent les structures de cache présentées à la section 3.3. Nous envisageons un premier paradigme d'édition directe des structures pour tester la faisabilité de notre proposition, et un second paradigme d'édition inverse qui facilitera la spécification du résultat. Contrairement à ce qui a été fait dans les systèmes de rendu inverse précédents, nous ne cherchons pas à résoudre une somme de contraintes globales. En effet, nous avons vu que les problèmes d'optimisation inverses non-linéaires deviennent rapidement trop complexes à résoudre. Il nous semble plus important de proposer une édition locale de l'éclairage indirect et de laisser ensuite le choix à l'utilisateur de propager ou non cette modification dans le reste de la scène.

## 3.5 Recherche proposée en édition

Le système d'édition que nous proposons repose sur l'utilisation des structures 2D ou 3D locales et d'une structure 3D globale à la scène qui ont mis en cache l'éclairage indirect. L'utilisateur modifie localement l'éclairage indirect et une fois cette modification effectuée, un processus de minimisation sous-contraintes recalcule les vecteurs d'irradiance pour donner l'effet souhaité. Il ne s'agit pas d'une minimisation globale à l'ensemble de la scène mais d'une minimisation locale à la zone de la structure éditée. Le nombre de structures de cache et leur type dépend des besoins de l'utilisateur. A savoir, si l'utilisateur ne veut effectuer que des modifications locales d'éclairage sur un objet, la structure de cache globale de la scène sera inutile. Remarquons que l'utilisateur n'édite pas la BRDF mais bien l'éclairage indirect. Il y a donc certaines modifications voulues par l'utilisateur qui ne pourront être satisfaites. Par exemple, forcer l'éclairage indirect à être de couleur rouge sur une surface qui ne réfléchit que du bleu ne donnera pas le résultat escompté, à savoir : obtenir une surface rouge. Dans ce type de situation, le paradigme d'édition de BRDF est alors plus approprié.

Enfin, l'édition commence à l'instant où l'utilisateur est satisfait des structures telles qu'elles ont été calculées. Il peut avant d'éditer les structures demander le raffinement de ces dernières. Nous pensons qu'il vaut mieux éviter de recalculer (*i.e.* réévaluer l'équation 3.8) à la volée certaines zones de structures pendant la phase d'édition afin d'éviter des changements trop brusques dans la reconstruction de l'irradiance.

Afin d'éditer l'éclairage indirect nous proposons deux paradigmes d'édition :

- Paradigme à modification directe. Dans ce paradigme l'utilisateur édite la représentation vectorielle directement. En fonction du voxel et d'un voisinage sélectionné, un ensemble de vecteurs est affiché et il peut modifier leur comportement (direction et norme cf. section 3.5.2). La structure se met alors à jour et les surfaces contenues dans le voxel et son voisinage voient leur apparence changer en conséquence.
- Paradigme orienté peinture. Dans ce paradigme l'utilisateur peint directement sur la surface dont il veut modifier l'éclairage indirect. C'est un paradigme d'édition inverse où l'utilisateur spécifie la couleur finale qu'il souhaite obtenir sur un point particulier de la surface. Le système tente ensuite de satisfaire les contraintes spécifiées.

### 3.5.1 Similarités des deux paradigmes

On peut considérer que le second paradigme est une extension du premier qui facilite l'interface d'édition pour l'utilisateur. Les deux paradigmes d'édition possèdent des similarités, à savoir :

- L'utilisateur doit définir la taille du voisinage de la structure sur lequel les modifications vont s'appliquer.
- La structure est raffinée lorsque sa résolution n'est pas suffisante pour pouvoir effectuer une modification locale.
- L'utilisateur peut choisir de propager le résultat de ses modifications locales au reste de la scène (cf. section 3.5.4)

Nous prévoyons des outils d'édition de la structure à portée locale et un outil à portée globale. Les outils d'édition locale reposent sur la modification des vecteurs d'irradiance en utilisant la séparabilité de leur représentation (cf. section 3.3.2). Nous envisageons également de proposer un outil global de lissage, qui permettrait de lisser l'ensemble de la structure. Cet outil est en fait une interface au mécanisme utilisé pour préconditionner de manière interne les structures (cf. section 3.3.2).

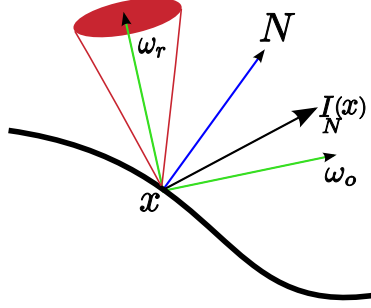


FIG. 3.9 – Edition de l'éclairage indirect dans le cas d'une BRDF *glossy*. Pour faciliter la tâche d'édition de l'utilisateur, le cône de directions du lobe spéculaire de la BRDF sous-jacente est affiché. L'utilisateur peut donc facilement modifier le vecteur d'irradiance  $I_N(x)$  et l'amener dans ce cône pour que son édition ait un impact sur l'apparence finale du point  $x$ .

Enfin, l'interface utilisateur proposée tiendra également compte de la BRDF des surfaces environnantes. Plus la BRDF est spéculaire, plus il est important de présenter à l'utilisateur le cône de directions depuis lequel la lumière incidente doit arriver pour qu'elle ait une influence dans l'apparence finale de la surface (cf. figure 3.9). Dans le premier paradigme d'édition, ce cône de directions peut être compliqué à afficher parce que le voxel peut contenir plusieurs objets (s'il n'y a pas d'objets dans le voxel la question ne se pose pas). Il faut donc définir un repère local  $(B, T, N)$  à l'ensemble de la géométrie contenue dans le voxel en utilisant une moyenne pondérée des normales. Dans le second paradigme d'édition, cela ne pose pas de problème, a priori, puisque la normale et la tangente à la surface ont été récupérées dans le pixel.

### 3.5.2 Paradigme d'édition directe

Dans le paradigme d'édition directe, l'usager modifie directement la structure en manipulant les vecteurs d'irradiance. Décrivons d'abord notre procédé de sélection des vecteurs et ensuite les modifications que nous permettrons à l'utilisateur d'effectuer.

L'utilisateur sélectionne une cellule de la structure en cliquant sur une surface. Le système affiche alors, au centre de la cellule sélectionnée, la base de vecteurs d'irradiance ainsi que la structure elle-même. Le voisinage de la structure est aussi mis en évidence. L'utilisateur peut également, en cliquant sur un sommet de la cellule ou de la grille, choisir de modifier une autre base de vecteurs d'irradiance. Dans le cas volumique, l'utilisateur peut déplacer le voxel sélectionné pour l'amener sur un voxel ne contenant aucune géométrie. La taille du voisinage est spécifiée par l'utilisateur de manière directe (via un menu ou une touche).

Une fois la base de vecteurs d'irradiance sélectionnée, nous proposons à l'utilisateur de manipuler la direction via la souris, la colorimétrie via une sélection sur une palette de couleur ou encore l'énergie des vecteurs d'irradiance. En effet, comme expliqué à la section 3.3.2, les vecteurs d'irradiance contiennent une information colorimétrique (équation (3.9)) et pour chacune de ses valeurs une direction (équation 3.10) y est associée. La force de la représentation est qu'il est possible de décorréler les informations colorimétriques et directionnelles. Les différents types de modification prévus sont :

- modification de la direction de chaque couleur, *i.e.* modifier  $\mathcal{D}_i(I_N)$
- spécification d'une direction globale pour le vecteur d'irradiance ; les directions par couleur sont changées en conséquence, *i.e.* minimiser un déplacement des  $\mathcal{D}_i(I_N)$  pour satisfaire la

- nouvelle direction demandée
- modification de chaque couleur sans changer les directions auxquelles elles sont associées, *i.e.* modifier une composante de  $\mathcal{C}(I_N)$
- modification de la norme des couleurs pour augmenter ou diminuer l'intensité du vecteur d'irradiance, *i.e.* modifier la normale de  $\mathcal{C}(I_N)$

Une fois la modification effectuée, le système tente de trouver la meilleure (celle qui minimise le changement) modification des vecteurs d'irradiance sur le voisinage choisi. Le fait d'avoir séparé les outils d'édition facilite le processus de minimisation.

Ce paradigme d'édition permet de modifier volumétriquement l'éclairage de l'ensemble d'un voxel et il autorise également la modification de l'éclairage dans un voxel vide. La manipulation de vecteurs et de couleurs simulent l'édition de sources de lumière directionnelles locales et constitue une interface d'édition qui nous semble simple. La sélection des vecteurs d'irradiance n'est pas nécessairement liée à la sélection d'un sommet de la structure ; il reste possible pour l'utilisateur de sélectionner un point d'une surface. Dans ce cas, le système lui présente les vecteurs d'irradiance interpolés à cet endroit. Le processus de minimisation est plus simple qu'en rendu inverse grâce à la séparation des outils, mais l'édition proposée ne permet pas de spécifier le résultat souhaité.

### 3.5.3 Paradigme d'édition orientée peinture

La seule différence avec le paradigme précédent réside dans le fait que l'utilisateur peint le résultat qu'il souhaite obtenir. Son édition se fait en sélectionnant un pinceau puis en l'appliquant sur la surface dont il souhaite modifier l'apparence. Toujours dans un souci de faciliter le processus de minimisation, l'utilisateur choisit parmi différents pinceaux ce qu'il voudrait modifier sur la surface de l'objet, à savoir :

- Un pinceau colorimétrique dont il a choisi la couleur via une palette de couleur
- Un pinceau énergétique qui n'agit que sur la luminosité.

Nous pensons que la direction ne peut être éditée que directement et le paradigme de peinture s'y applique donc mal. Dès lors, le système de minimisation peut jouer sur les directions des vecteurs d'irradiance pour satisfaire les contraintes de peinture.

La difficulté pour mettre en place ce paradigme se posera au niveau de la minimisation. En effet, puisque l'utilisateur édite sur une surface et donc pour une normale sous-jacente, il y a non seulement un processus de minimisation sur l'interpolation directionnelle mais aussi sur l'interpolation spatiale. L'interpolation directionnelle vient du fait qu'il a fallu interpoler (cf. section 3.3.3) le vecteur d'irradiance pour la normale au point où l'utilisateur a cliqué. L'interpolation spatiale est identique au paradigme d'édition précédent. Une modification des vecteurs d'irradiance contenus dans le voisinage sélectionné est nécessaire pour obtenir l'effet souhaité. En revanche, la facilité à spécifier le résultat directement sur une surface (plutôt que dans un volume) nous semble être un net avantage comparativement au paradigme précédent.

### 3.5.4 Interaction après édition

Dans le système que nous envisageons, une fois l'édition locale effectuée, l'utilisateur, peut propager sa modification sur l'environnement. Pour ce faire, une structure volumétrique pour l'ensemble de la scène doit avoir été précalculée. Cette structure fait office de fonction de transfert.

Puisque les structures locales intersectent la structure globale, qui elle englobe l'ensemble du volume de la scène, il est possible de modifier les vecteurs d'irradiance de la structure globale par le

même type de processus de minimisation présenté précédemment. En effet, les vecteurs d'irradiance locaux sont contenus dans un nombre fini de cellules de la structure globale. Une fois ces vecteurs globaux modifiés, nous pouvons répercuter sur d'autres structures locales les modifications qu'ils ont subies.

La portée de la modification effectuée par l'utilisateur est entièrement paramétrable. Elle peut être :

- globale : la structure globale se met à jour entièrement à l'aide d'un processus de diffusion par exemple, et répercute sur l'ensemble des structures locales les modifications de l'utilisateur.
- localisée par objet : la structure globale ne reporte les modifications que pour les objets sélectionnés par l'utilisateur.

Une autre forme d'interaction possible est de laisser l'utilisateur déplacer un objet dans la scène. Il peut alors choisir entre influencer la structure globale ou recevoir l'influence de cette dernière.



## Chapitre 4

# Échéancier

Jusqu'à présent le doctorat s'est déroulé comme suit :

Session	Lieu	Travail
Automne 2005	Bordeaux	Recherche et développement de la technique de <i>キャッシング</i>
Hiver 2006	Bordeaux	Recherche et développement de la technique de <i>キャッシング</i>
Été 2006	Montréal	Travail prospectif sur l'édition des phénomènes d'éclairage
Automne 2006	Montréal	Cours IFT6042 Synthèse d'images et prédoc I
Hiver 2007	Bordeaux	Recherche et Développement de l'édition vectorielle de BRDF
Été 2007	Bordeaux-Montréal	Travail d'analyse sur la technique de <i>キャッシング</i> et rédaction du prédoc II
Automne 2007	Montréal-Bordeaux	Cours d'imagerie médicale à l'Université Bordeaux et Recherche et soumission à I3D
Hiver 2008	Bordeaux	Recherche et soumission à EGSR
Été 2008	Bordeaux	Début de rédaction de la thèse et soumissions à EGWNP et CGF
Automne 2008	Bordeaux	Fin de rédaction et soutenance et soumission à EG

Plus spécifiquement notre plan de travail de recherche et de rédaction en fonction des soumissions aux forums est :

## Symposium on Interactive 3D Graphics

- Date de soumission : 22 octobre 2007
- Partie soumise : Édition vectorielle de BRDF
- Travail à fournir :
  - Interpolation sur le triangle sphérique
  - Interface pour permettre à l'utilisateur d'ajouter des configurations
  - Interface pour permettre à l'utilisateur de définir sa *free-form curve*
- Évaluation de la durée de travail : 1 mois

## *Eurographics Symposium on Rendering*

- Date de soumission : mi-avril 2008
- Partie soumise : Édition à l'aide du paradigme de peinture de BRDF
- Travail à fournir :
  - Développement de l'interface peinture
  - Remplissage et interpolation de la structure de BRDF creuse
  - Tests des différentes paramétrisations
- Évaluation de la durée de travail : 4 mois

## *Eurographics Workshop on Natural Phenomena* (conditionnel)

- Date de soumission : mi-mai 2008
- Partie soumise : Application de la grille aux arbres et arbustes, contexte interactif et qualité
- Travail à fournir :
  - préconditionnement aux frontières
- Évaluation de la durée de travail : 1 mois

## *Computer Graphics Forum*

- Date de soumission : août 2008
- Partie soumise : article d'analyse sur le fonctionnement de la grille en tant que structure de cache et comparaisons avec les méthodes existantes
- Travail à fournir :
  - préconditionnement interne
  - comparaisons avec les harmoniques sphériques
  - comparaisons avec les techniques de Tabellion et al. [TL04] et Ward et al [WH92].
- Évaluation de la durée de travail : 2 mois

## *Eurographics*

- Date de soumission : fin septembre 2009
- Partie soumise : Edition de l'éclairage indirect
- Travail à fournir :
  - interface pour l'édition
  - mini système de minimisation, *i.e.* de résolution inverse
  - transferts entre les différentes structures (globales ↔ locales)
- Évaluation de la durée de travail : 1 mois et demi



## Chapitre 5

# Conclusion et travaux futurs

### 5.1 Conclusion

Le gain de puissance constant des CPUs, l'accès à des giga-octets de mémoire vive, et l'introduction de la partie programmable pour les GPUs ont permis à la synthèse d'images d'atteindre des niveaux de photo-réalisme très élevés. L'amélioration de la qualité photo-réaliste des images s'explique aussi par une étude toujours plus poussée des BRDFs et des algorithmes d'éclairage global. Cependant, les fonctions et les algorithmes dont résulte cette qualité sont complexes à contrôler pour un artiste qui n'est ni un physicien, ni un mathématicien et ni un informaticien. En effet, le nombre et la dimensionnalité des paramètres en cause sont élevés et de plus, ils possèdent une interdépendance non-linéaire sur l'apparence finale. Ainsi contrôler l'apparence finale des scènes reste un travail pénible, d'essais et erreurs souvent peu intuitifs et frustrants surtout lorsque le résultat de chaque modification requière plusieurs secondes, voire minutes et même heures dans le cas de d'éclairage global.

Notre recherche vise à fournir des outils interactifs d'édition pour les BRDFs et l'éclairage global adaptés aux besoins et abstractions des artistes. Le concept central de notre recherche est de relâcher certaines contraintes liées à la physique pour offrir plus de flexibilité et d'expressivité artistique tout en assurant de garder un certain degré de cohérence lorsque la caméra, les objets ou les sources de lumière sont déplacées. Dès lors, nous avons proposé deux paradigmes d'édition (directe et inverse) pour deux phénomènes d'éclairage omniprésents en synthèse d'images.

### 5.2 Travaux futurs

Dues aux limitations de temps imposées au doctorat, certaines réalisations non-abordées dans ce document ne pourront être complétées. Nous indiquons ici certaines directions qui méritent une réflexion plus approfondie.

Dans le cadre de l'édition de BRDF, nous voudrions étudier comment rendre la BRDF physiquement réaliste en respectant les contraintes de réciprocité et de conservation de l'énergie tout en affectant le moins possible les modifications appliquées par l'artiste. De plus, l'unification des paradigmes d'édition vectorielle et de peinture nous semble une piste de recherche intéressante en puisant selon l'expérience que nous aurons acquise dans les avantages de chacun.

Sur l'édition de l'éclairage indirect, il serait également intéressant d'approfondir la possibilité d'inclure des contraintes physiques dans le processus inverse. Nous aimerions autoriser l'utilisateur

à spécifier des contraintes d'éclairage directionnel très fortes. Pour ce faire, nous envisageons de réaligner la base de vecteur d'irradiance avec la direction principale d'éclairage définie par l'artiste ou encore de dériver une nouvelle base de vecteurs d'irradiance représentant plus de six directions.

Le développement de structures pour l'édition d'autres phénomènes d'éclairage tels que les ombres, les caustiques et les BTFs serait tout aussi intéressant. De plus, l'unification des structures pour tout type de phénomènes d'éclairage serait une extension importante, si une telle superstructure répond adéquatement aux besoins artistiques.

L'édition dans d'autres domaines tels que l'imagerie médicale, la visualisation de données ou encore l'archéologie constitue aussi une voie intéressante. Ces domaines ont également un besoin d'édition, mais le contexte et les problématiques sont différentes de celles abordées dans cette proposition. En effet, ce n'est pas l'artiste qui est au centre de l'édition, mais un spécialiste qui doit interpréter correctement des images.

Finalement, tous ces outils et techniques d'édition sont développés à l'intention des utilisateurs. Bien que nous nous sommes inspirés de notre propre expérience et de discussions avec de tels utilisateurs, il serait important de réaliser une étude approfondie sur des spécialistes en production d'images de synthèse afin d'évaluer les améliorations obtenues, les difficultés rencontrées et leur retour d'expérience.

# Bibliographie

- [AFO05] O. Arikian, D.A. Forsyth et J.F. O'Brien. « Fast and Detailed Approximate Global Illumination by Irradiance Decomposition ». Dans *Proc. SIGGRAPH 2005*, pages 1108–1114. ACM, 2005.
- [Aga07] Aseem Agarwala. « Efficient gradient-domain compositing using quadtrees ». *ACM Trans. Graph.*, volume 26, numéro 3, page 516, 2007.
- [AKFR07] Ben-Artzi Aner, Egan Kevin, Durand Fredo et Ramamoorthi Ravi. « A Precomputed Polynomial Representation for Interactive BRDF Editing with Global Illumination ». *To appear in ACM Trans. Graph.*, 2007.
- [APS00] M. Ashikhmin, S. Premoze et P. Shirley. « A microfacet-based BRDF generator ». Dans *ACM SIGGRAPH '00*, pages 65–74, 2000.
- [Arv94] J. Arvo. « The irradiance Jacobian for partially occluded polyhedral sources ». Dans *Proc. SIGGRAPH'94*, pages 343–350. ACM, 1994.
- [AS00] M. Ashikhmin et P. Shirley. « An anisotropic phong light reflection model », 2000.
- [AWB06] Ken-Ichi Anjyo, Shuhei Wemler et William Baxter. « Tweakable light and shade for cartoon animation ». Dans *NPAC '06 : Proc. international symposium on Non-photorealistic animation and rendering*, pages 133–139. ACM Press, 2006.
- [BAOR06] Aner Ben-Artzi, Ryan Overbeck et Ravi Ramamoorthi. « Real-time BRDF editing in complex lighting ». *ACM Trans. Graph.*, volume 25, numéro 3, pages 945–954, 2006.
- [Bar97] Ronen Barzel. « Lighting Controls for Computer Cinematography ». *journal of graphics tools*, volume 2, numéro 1, pages 1–20, 1997.
- [BFGS03] Jeff Bolz, Ian Farmer, Eitan Grinspun et Peter Schröder. « Sparse matrix solvers on the GPU : conjugate gradients and multigrid ». *ACM Trans. Graph.*, volume 22, numéro 3, pages 917–924, 2003.
- [Bli77] James F. Blinn. « Models of light reflection for computer synthesized pictures ». Dans *ACM SIGGRAPH '77*, pages 192–198, 1977.
- [CBP07] Luc Claustres, Loïc Barthe et Mathias Paulin. « Wavelet encoding of BRDFs for real-time rendering ». Dans *GI '07 : Proceedings of Graphics Interface 2007*, pages 169–176, New York, NY, USA, 2007. ACM Press.
- [CBS96] Benoît Crespin, Carole Blanc et Christophe Schlick. « Implicit Sweep Objects ». *Computer Graphics Forum*, volume 15, numéro 3, pages 165–174, 1996.
- [Chr99] P.H. Christensen. « Faster Global Photon Map Global Illumination ». *Journal of Graphics Tools*, volume 4, numéro 3, pages 1–10, 1999.

- [CMS87] Brian Cabral, Nelson Max et Rebecca Springmeyer. « Bidirectional reflection functions from surface bump maps ». Dans *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 273–281, New York, NY, USA, 1987. ACM Press.
- [CPK06] M. Colbert, S. Pattanaik et J. Krivánek. « BRDF-Shop : Creating Physically Correct Bidirectional Reflectance Distribution Functions ». *IEEE Computer Graphics and Applications*, volume 26, numéro 1, pages 30–36, 2006.
- [CSF99] António Cardoso Costa, António Augusto Sousa et Fernando Nunes Ferreira. « Lighting Design : A Goal Based Approach using Optimisation. ». Dans *Rendering Techniques*, pages 317–328, 1999.
- [CT82] R. L. Cook et K. E. Torrance. « A Reflectance Model for Computer Graphics ». *ACM Trans. Graph.*, volume 1, numéro 1, pages 7–24, 1982.
- [DBB06] Philip Dutré, Kavita Bala et Philippe Bekaert. *Advanced Global Illumination*. A. K. Peters, Ltd., 2006.
- [DD00] Frédo Durand et Julie Dorsey. « Interactive Tone Mapping ». 2000. Held in Brno, Czech Republic.
- [DHDS05] Florent Duguet, Carlos Hernandez, George Drettakis et Francis Schmitt. « Level of Detail Continuum for Huge Geometric Data ». Rapport technique RR-5552, INRIA, REVES/INRIA Sophia-Antipolis, April 2005.
- [ES04] Carlos Hernández Esteban et Francis Schmitt. « Silhouette and stereo fusion for 3D object modeling ». *Comput. Vis. Image Underst.*, volume 96, numéro 3, pages 367–392, 2004.
- [GD01] Xavier Granier et George Drettakis. « Incremental Updates for Rapid Glossy Global Illumination ». Dans *Proceedings of Eurographics 2001 conference*, pages 268–277. Eurographics, Blackwell Publishing, 2001.
- [GH00] Reid Gershbein et Pat Hanrahan. « A fast relighting engine for interactive cinematic lighting design ». Dans *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 353–358, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [GSHG98] G. Greger, P. Shirley, P.M. Hubbard et D.P. Greenberg. « The Irradiance Volume ». *IEEE Computer Graphics and Applications*, volume 18, numéro 2, pages 32–43, 1998.
- [Hac05] Toshiya Hachisuka. « High-Quality Global Illumination Rendering Using Rasterization ». Dans *GPU Gems 2 : Programming Techniques for High Performance Graphics and General-Purpose Computation*, pages 591–614. 2005.
- [HHP<sup>+</sup>92] Xiao D. He, Patrick O. Heynen, Richard L. Phillips, Kenneth E. Torrance, David H. Salesin et Donald P. Greenberg. « A fast and accurate light reflection model ». *SIGGRAPH Comput. Graph.*, volume 26, numéro 2, pages 253–254, 1992.
- [HPB06] Miloš Hašan, Fabio Pellacini et Kavita Bala. « Direct-to-indirect transfer for cinematic relighting ». *ACM Trans. Graph.*, volume 25, numéro 3, pages 1089–1097, 2006.
- [HTSG91] Xiao D. He, Kenneth E. Torrance, François X. Sillion et Donald P. Greenberg. « A comprehensive physical model for light reflection ». Dans *ACM SIGGRAPH '91*, pages 175–186, 1991.

- [htt05] <http://www.chaosgroup.com/>. « The Chaos Group ». Internet Ressource, 2005 ?
- [Jen96] Henrik Wann Jensen. « Global illumination using photon maps ». Dans *Proceedings of the eurographics workshop on Rendering techniques '96*, pages 21–30, London, UK, 1996. Springer-Verlag.
- [Jen01] Henrik Wann Jensen. *Realistic Image Synthesis using Photon Mapping*. A.K. Peters, 2001.
- [Kaj85] James T. Kajiya. « Anisotropic reflection models ». Dans *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 15–21, New York, NY, USA, 1985. ACM Press.
- [Kaj86] J.T. Kajiya. « The rendering equation ». Dans *Proc. SIGGRAPH'86*, pages 143–150, 1986.
- [KAMJ05] Anders Wang Kristensen, Tomas Akenine-Möller et Henrik Wann Jensen. « Precomputed local radiance transfer for real-time lighting design ». Dans *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 1208–1215, New York, NY, USA, 2005. ACM Press.
- [KBPv06] J. Křivánek, K. Bouatouch, S.N. Pattanaik et J. Žára. « Making Radiance and Irradiance Caching Practical : Adaptive Caching and Neighbor Clamping ». Dans *Proc. Eurographics Symposium on Rendering 2006*, pages 127–138, 2006.
- [KBPZ06] J. Křivánek, Kadi Bouatouch, Sumanta Pattanaik et Jiri Žára. « Making Radiance and Irradiance Caching Practical : Adaptive Caching and Neighbor Clamping ». *Symposium on Rendering*, volume 17, pages 127–138, 2006.
- [KGPB05] J. Křivánek, P. Gautron, S. Pattanaik et K. Bouatouch. « Radiance caching for efficient global illumination computation ». *IEEE Trans. Visualization and Computer Graphics*, volume 11, numéro 5, pages 550–561, 2005.
- [KKB<sup>+</sup>05] Jaroslav Křivánek, Jaakko Konttinen, Kadi Bouatouch, Sumanta Pattanaik et Jiří Žára. « Fast Approximation to Spherical Harmonic Rotation ». Dans *SCCG '06 : Proc. spring conference on Computer graphics*. ACM Press, 2005.
- [KM99a] Jan Kautz et Michael D. McCool. « Interactive Rendering with Arbitrary BRDFs using Separable Approximations ». Dans *Proc. Eurographics Workshop on Rendering*, pages 247–260. Springer, juin 1999.
- [KM99b] Jan Kautz et Michael D. McCool. « Interactive rendering with arbitrary BRDFs using separable approximations ». Dans *SIGGRAPH '99 : ACM SIGGRAPH 99 Conference abstracts and applications*, page 253, 1999.
- [KM00] Jan Kautz et Michael D. McCool. « Approximation of Glossy Reflection with Prefiltered Environment Maps ». Dans *Graphics Interface*, pages 119–126, 2000.
- [KPC93] John K. Kawai, James S. Painter et Michael F. Cohen. « Radioptimization : goal based rendering ». Dans *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 147–154, New York, NY, USA, 1993. ACM Press.
- [LBAD<sup>+</sup>06] Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi et Szymon Rusinkiewicz. « Inverse shade trees for non-parametric material representation and editing ». *ACM Trans. Graph.*, volume 25, numéro 3, pages 735–745, 2006.

- [LFTG97] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance et Donald P. Greenberg. « Non-linear approximation of reflectance functions ». Dans *ACM SIGGRAPH '97*, pages 117–126, 1997.
- [LFTW05] Hongsong Li, Sing Choong Foo, Kenneth E. Torrance et Stephen H. Westin. « Automated three-axis gonioreflectometer for computer graphics applications ». Dans A. Duparre, B. Singh et Z-H Gu, éditeurs. *Advanced Characterization Techniques for Optics, Semiconductors, and Nanotechnologies II*, volume Proceedings of SPIE Vol. 5878, pages 5878–29. SPIE, SPIE, Bellingham, WA, juillet 2005.
- [LSK<sup>+</sup>06] Aaron E. Lefohn, Shubhabrata Sengupta, Joe Kniss, Robert Strzodka et John D. Owens. « Glift : Generic, efficient, random-access GPU data structures ». *ACM Trans. Graph.*, volume 25, numéro 1, pages 60–99, 2006.
- [LW93] E.P. Lafortune et Y.D. Willems. « Bi-directional Path Tracing ». Dans *Proc. Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, 1993.
- [LW94] Eric P. Lafortune et Yves D. Willems. « Using the Modified Phong BRDF for Physically Based Rendering ». Rapport technique CW197, Katholieke Universiteit Leuven, Belgium, 1994.
- [MAA01] Michael D. McCool, Jason Ang et Anis Ahmad. « Homomorphic factorization of BRDFs for high-performance rendering ». Dans *ACM SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 171–178. ACM Press, 2001.
- [MMS<sup>+</sup>06] G. Müller, J. Meseth, M. Sattler, R. Sarlette et R. Klein. « Acquisition, Synthesis, and Rendering of Bidirectional Texture Functions ». *Computer Graphics Forum*, volume 24, numéro 1, pages 83–109, 2006.
- [MPBM03] W. Matusik, H. Pfister, M. Brand et L. McMillan. « A Data-Driven Reflectance Model ». *ACM Transactions on Graphics*, volume 22, numéro 3, pages 759–769, juillet 2003.
- [NDM05] A. Ngan, F. Durand et W. Matusik. « Experimental Analysis of BRDF Models ». Dans *Proc. Eurographics Symposium on Rendering*, pages 117–226, 2005.
- [NPG05] Mangesh Nijasure, Sumanta N. Pattanaik et Vineet Goel. « Real-Time Global Illumination on GPUs ». *J. Graph. Tools*, volume 10, numéro 2, pages 55–71, 2005.
- [OMS<sup>+</sup>07] Makoto Okabe, Yasuyuki Matsushita, Li Shen, et Takeo Igarashi. « Illumination Brush : Interactive Design of All-frequency Lighting ». Dans *To appear in the Proceedings of Pacific Graphics*, 2007.
- [ON94] Michael Oren et Shree K. Nayar. « Generalization of Lambert's reflectance model ». Dans *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246, New York, NY, USA, 1994. ACM Press.
- [PBMF07] Fabio Pellacini, Frank Battaglia, R. Keith Morley et Adam Finkelstein. « Lighting with paint ». *ACM Trans. Graph.*, volume 26, numéro 2, page 9, e 07.
- [PF90] Pierre Poulin et Alain Fournier. « A Model for Anisotropic Reflection ». *Computer Graphics*, volume 24, numéro 4, pages 273–282, août 1990.
- [PF92] Pierre Poulin et Alain Fournier. « Lights from highlights and shadows ». *Computer Graphics*, volume 25, numéro 2, pages 31–38, mars 1992.

- [PF95] P. Poulin et A. Fournier. « Painting Surface Characteristics ». Dans *Proc. Eurographics Workshop on Rendering*, pages 160–169. Springer, juin 1995.
- [PGB03] Patrick Pérez, Michel Gangnet et Andrew Blake. « Poisson image editing ». Dans *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers*, pages 313–318, New York, NY, USA, 2003. ACM Press.
- [Pho75] Bui Tuong Phong. « Illumination for computer generated pictures ». *Commun. ACM*, volume 18, numéro 6, pages 311–317, 1975.
- [Pop03] Stéphane Popinet. « Gerris : a tree-based adaptive solver for the incompressible Euler equations in complex geometries ». *J. Comput. Phys.*, volume 190, numéro 2, pages 572–600, 2003.
- [PRJ97] Pierre Poulin, Karim Ratib et Marco Jacques. « Sketching Shadows and Highlights to Position Lights ». Dans *Proceedings of Computer Graphics International 97*, pages 56–63. IEEE Computer Society, juin 1997.
- [PVL<sup>+</sup>05] Fabio Pellacini, Kiril Vidimčec, Aaron Lefohn, Alex Mohr, Mark Leone et John Warren. « Lpics : a hybrid hardware-accelerated relighting engine for computer cinematography ». Dans *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 464–470, New York, NY, USA, 2005. ACM Press.
- [RH01] Ravi Ramamoorthi et Pat Hanrahan. « A signal-processing framework for inverse rendering ». Dans *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 117–128, New York, NY, USA, 2001. ACM Press.
- [Rus98] Szymon Rusinkiewicz. « A New Change of Variables for Efficient BRDF Representation ». Dans *Proc. Eurographics Rendering Workshop*, pages 11–22. Springer, 1998.
- [Sch94a] Christophe Schlick. « An Inexpensive BRDF Model for Physically-Based Rendering ». *Computer Graphics Forum*, volume 13, numéro 3, pages 233–246, 1994.
- [Sch94b] Christophe Schlick. « A fast alternative to Phong's specular model ». pages 385–387, 1994.
- [Sch94c] Christophe Schlick. « Quantization Techniques for Visualization of High Dynamic Range Pictures ». pages 7–18, 1994.
- [SDS<sup>+</sup>93] Chris Schoeneman, Julie Dorsey, Brian Smits, James Arvo et Donald Greenburg. « Painting with light ». Dans *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 143–146, New York, NY, USA, 1993. ACM Press.
- [SMGG01] Peter-Pike Sloan, William Martin, Amy Gooch et Bruce Gooch. « The Lit Sphere : A Model for Capturing NPR Shading from Art ». Dans *Proc. Graphics Interface*, pages 143–150, 2001.
- [SP94] François Xavier Sillion et Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, 1994.
- [SZC<sup>+</sup>07] Xin Sun, Kun Zhou, Yanyun Chen, Stephen Lin, Jiaoying Shi et Baining Guo. « Interactive relighting with dynamic BRDFs ». *ACM Trans. Graph.*, volume 26, numéro 3, page 27, 2007.

- [TL04] E. Tabellion et A. Lamorlette. « An Approximate Global Illumination System for Computer Generated Films ». *ACM Trans. on Graphics*, volume 23, numéro 3, pages 469–476, 2004.
- [TPWG02] Parag Tole, Fabio Pellacini, Bruce Walter et Donald P. Greenberg. « Interactive global illumination in dynamic scenes ». Dans *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 537–546, New York, NY, USA, 2002. ACM Press.
- [TR93] Jack Tumblin et Holly Rushmeier. « Tone Reproduction for Realistic Images ». *IEEE Comput. Graph. Appl.*, volume 13, numéro 6, pages 42–48, 1993.
- [TS67] K. E. Torrance et E. M. Sparrow. « Theory for off-specular reflection from roughened surfaces ». pages 32–41, 1967.
- [VG94] Eric Veach et Leonidas J. Guibas. « Bidirectional Estimators for Light Transport ». Dans *Proc. Eurographics Workshop Rendering*, pages 147–162, 1994.
- [VG97] E. Veach et L.J. Guibas. « Metropolis light transport ». Dans *Proc. SIGGRAPH'97*, pages 65–76, 1997.
- [War91] Gregory Ward. « Real Pixels ». Dans *Graphics Gems II*, pages 80–83. 1991.
- [War92] Gregory J. Ward. « Measuring and modeling anisotropic reflection ». Dans *ACM SIGGRAPH '92*, pages 265–272, 1992.
- [War94] Greg Ward. « A contrast-based scalefactor for luminance display ». pages 415–421, 1994.
- [WAT92] Stephen H. Westin, James R. Arvo et Kenneth E. Torrance. « Predicting reflectance functions from complex surfaces ». *SIGGRAPH Comput. Graph.*, volume 26, numéro 2, pages 255–264, 1992.
- [WH92] Gregory J. Ward et Paul Heckbert. « Irradiance Gradients ». Dans *Proc. Eurographics Workshop on Rendering 1992*, pages 85–98, 1992.
- [Whi80] Turner Whitted. « An improved illumination model for shaded display ». Dans *Communications of the ACM*, 23(6), June 1980.
- [WRC88] G.J. Ward, F.M. Rubinstein et R.D. Clear. « A ray tracing solution for diffuse inter-reflection ». Dans *Proc. SIGGRAPH'88*, pages 85–92, 1988.
- [WTL04] Rui Wang, John Tran et David Luebke. « All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation ». Dans *Proc. Eurographics Symposium on Rendering 2004*, pages 321–330, 2004.
- [XP98] C. Xu et J. Prince. « Snakes, Shapes, and Gradient Vector Flow ». *IEEE Transactions on Image Processing*, volume 7, pages 359–369, 1998.
- [ZSP98] J. Zaninetti, X. Serpaggi et B. Péroche. « A Vector Approach for Global Illumination in Ray Tracing ». Dans *Proc. Eurographics 1998*, pages 149–158, 1998.



# Annexe A

## Radiométrie et Photométrie

La radiométrie est l'étude de la lumière et plus particulièrement de sa mesure physique sur l'ensemble du spectre. La photométrie qui en découle prend en compte la perception de l'énergie lumineuse par le système visuel humain qui est uniquement capable d'observer les phénomènes compris approximativement entre 380 et 780 nm.

On trouvera dans la table A.5 une correspondance entre les quantités et unités radiométriques et photométriques.

### A.1 Quantités radiométriques

#### Flux énergétique spectral ou puissance spectrale

La quantité de lumière émanant/quittant/traversant une surface est appelée énergie rayonnante spectrale (cf. annexe A.2). Si l'on considère la variation de cette dernière par rapport au temps, on établit le flux énergétique spectral<sup>1</sup>  $\Phi_\lambda$  dont le concept est illustré à la figure A.1(a).

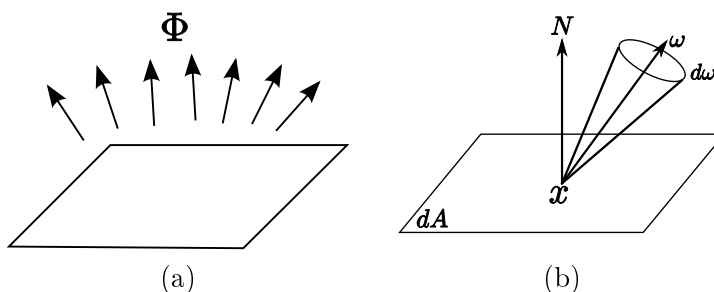


FIG. A.1 – (a) Notion de flux. (b) Repérage de la radiance. La radiance est un scalaire valable pour un angle solide  $d\omega$  repéré par une direction  $\omega$ .

En synthèse d'images, on considère que le système a atteint un état énergétique lumineux stable et l'aspect temporel du flux est donc ignoré.

<sup>1</sup>La littérature anglo-saxonne parle de *spectral radiant flux* en ce qui concerne le flux énergétique spectral et de *spectral radiant power* pour la puissance énergétique.

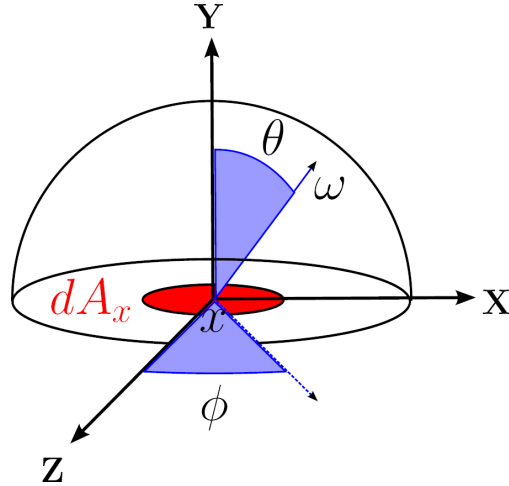


FIG. A.2 – Reperage hémisphérique des directions.

### Irradiance et radiosité spectrales

Les quantités permettant de calculer l'énergie lumineuse qui part ou qui arrive sur une surface donnée sont l'irradiance et la radiosité spectrales. Si l'on ne considère pas la provenance de l'énergie, on parle de densité de flux énergétique.

L'irradiance<sup>2</sup> spectrale,  $E_\lambda(x)$ , en un point  $x$ , exprimée en Watt par mètre au carré par nanomètre ( $Wm^{-2}nm^{-1}$ ) correspond au flux énergétique spectral incident sur une surface par aire de surface unitaire :

$$E_\lambda(x) = \frac{d\Phi_\lambda}{dA} \quad (\text{A.1})$$

avec  $dA$  surface unitaire élémentaire.

La radiosité spectrale  $B_\lambda$ , parfois appelée excitation énergétique, correspond au flux énergétique partant d'une surface par aire de surface unitaire. Tout comme l'irradiance, elle est aussi exprimée en Watt par mètre au carré par nanomètre ( $Wm^{-2}nm^{-1}$ ). Elle est définie par l'équation suivante :

$$B_\lambda(x) = \frac{d\Phi_\lambda}{dA}. \quad (\text{A.2})$$

### Notion d'angle solide et d'angle solide différentiel

Dans ce document, les coordonnées sphériques sont repérées par les angles  $\theta$  et  $\phi$ , où  $\theta$  est à valeur sur l'intervalle  $[0, \frac{\pi}{2}]$  et représente l'angle co-latitudinal, tandis que  $\phi$  est à valeur sur l'intervalle  $[0, 2\pi]$  et représente l'angle azimutal (longitudinal), comme illustré à la figure A.2.

Une quantité fondamentale en éclairage global associée à la notion de direction est l'angle solide (cf. figure A.3) qui permet d'exprimer à la fois une portion de direction et une aire infinitésimale.

L'angle solide  $\Omega$  au point  $x$ , défini par une aire  $A$  sur la sphère est défini par :

$$\Omega_x = \frac{A}{r^2} \quad (\text{A.3})$$

avec  $r$  le rayon de la sphère.

Bien que sans dimension, les angles solides sont exprimés en stéradian ( $sr$ ). Pour calculer l'angle solide d'un objet quelconque, on doit projeter la surface de cet objet sur la sphère unité.

<sup>2</sup>Le terme irradiance est un anglicisme courant. Le terme français officiel est « éclairage énergétique ».

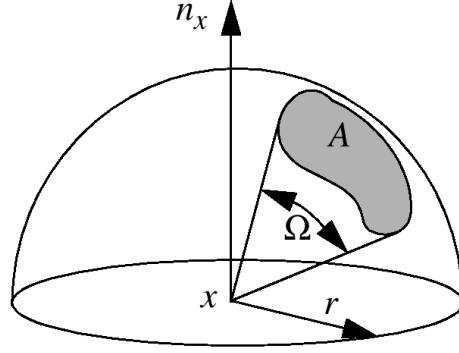


FIG. A.3 – Angle solide subtendu par une surface  $A$  au point  $x$ .

Pour pouvoir intégrer des fonctions sur l'hémisphère, on définit l'angle solide différentiel, dont une illustration est donnée en figure A.4(a), centré autour d'une direction  $\omega = (\theta, \phi)$ , notée  $d\omega$ , par :

$$d\omega = d\phi d\theta \sin \theta. \quad (\text{A.4})$$

Comme le montre l'image (a) de la figure A.4, le facteur  $\sin \theta$  s'explique par le fait que la variation d'angle solide est plus grande à l'horizon qu'au pôle.

Pour pouvoir passer d'une formulation hémisphérique à surfacique, on a besoin d'exprimer l'élément différentiel d'angle solide en fonction de l'élément différentiel de surface :

$$d\omega_x = \frac{\cos \theta_y dA_y}{d_{xy}^2} \quad (\text{A.5})$$

avec

- $\theta_y$  angle entre la normale au point  $y$  et  $\omega_x$
- $d_{xy}$  la distance euclidienne entre  $x$  et  $y$
- $dA_y$  l'élément différentiel de surface au point  $y$

dont une illustration est donnée à la figure A.4 en image (b).

Grâce à la notion d'angle solide, on peut alors définir de nouvelles quantités radiométriques qui tiennent compte de la direction de la lumière.

### Intensité énergétique spectrale et radiance spectrale

L'intensité énergétique spectrale<sup>3</sup>  $I_\lambda(\omega)$ , exprimée en Watt par stéradian ( $Wsr^{-1}$ ), correspond au flux énergétique par angle solide unitaire :

$$I_\lambda(\omega) = \frac{d\Phi_\lambda}{d\omega}. \quad (\text{A.6})$$

La radiance<sup>4</sup> spectrale  $L_\lambda$ , exprimée en Watt par mètre carré par stéradian par nanomètre ( $Wm^{-2}sr^{-1}nm^{-1}$ ), est définie par le flux énergétique spectral par surface unitaire projetée et par angle solide différentiel :

$$L_\lambda = \frac{d^2\Phi_\lambda}{\cos \theta dA d\omega}$$

<sup>3</sup> *Spectral radiant intensity* dans la littérature anglo-saxonne.

<sup>4</sup> Il s'agit une fois encore d'un anglicisme courant dans la communauté scientifique de l'éclairage global. Le terme français adéquat est « luminance énergétique ».

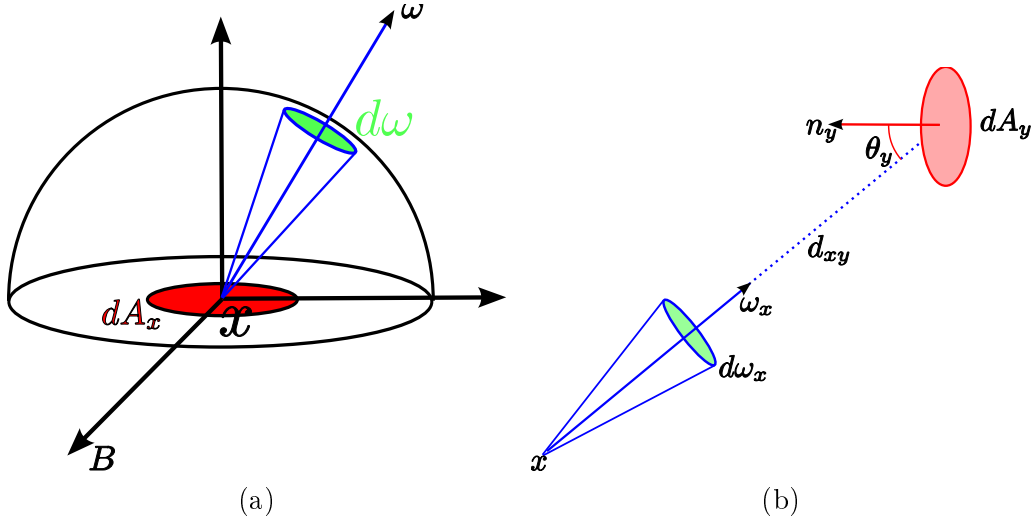


FIG. A.4 – (a) Angle solide différentiel  $d\omega$  pour une direction  $\omega$ . (b) Quantités en jeu pour la formulation surfacique de l'angle solide différentiel  $d\omega_x$ .

avec  $\theta$  angle entre la direction  $\omega$  et la normale de la surface.

Afin de différencier la radiance incidente de la radiance sortante, on utilise la notation suivante :

- $L_\lambda(x \rightarrow \omega)$  représente la radiance quittant le point  $x$  dans la direction  $\omega$ .
- $L_\lambda(x \leftarrow \omega)$  représente la radiance incidente au point  $x$  de direction  $\omega$ .

L'une des propriétés fondamentales de la radiance est son invariance dans le vide le long d'un rayon, c'est-à-dire :

$$L_\lambda(x \rightarrow y) = L_\lambda(y \leftarrow x). \quad (\text{A.7})$$

La radiance spectrale est sans conteste la quantité fondamentale en éclairage global. Notre système visuel comme les capteurs y sont sensibles. Malheureusement, c'est une quantité complexe puisque c'est une fonction de dimension 6 (3 pour la position, 2 pour la direction et 1 pour la longueur d'onde).

### Equations reliant les diverses quantités radiométriques

Les quantités radiométriques peuvent s'exprimer en fonction de la radiance spectrale :

$$\Phi_\lambda = \int_A \int_\Omega L_\lambda(x \rightarrow \omega) \cos \theta \, d\omega \, dA_x \quad (\text{A.8})$$

$$E_\lambda(x) = \int_\Omega L_\lambda(x \leftarrow \omega) \cos \theta \, d\omega \quad (\text{A.9})$$

$$B_\lambda(x) = \int_\Omega L_\lambda(x \rightarrow \omega) \cos \theta \, d\omega \quad (\text{A.10})$$

avec :

- $\Phi_\lambda$  le flux énergétique spectral
- $E_\lambda(x)$  l'irradiance spectrale
- $B_\lambda(x)$  la radiosité spectrale
- $L_\lambda$  la radiance spectrale.

<sup>5</sup>Il s'agit aussi d'un anglicisme. Le terme français approprié est « luminance lumineuse ».

Quantité radiométrique	Unité	Quantité photométrique	Unité
Flux énergétique	$W$	Flux lumineux	$lm$
Irradiance	$W \cdot m^{-2}$	Illuminance	$lm \cdot m^{-2}$ i.e. $lx$
Intensité énergétique	$W \cdot sr^{-1}$	Intensité lumineuse	$lm \cdot sr^{-1}$ i.e. $cd$
Radiance	$W \cdot m^{-2} \cdot sr^{-1}$	Luminance <sup>5</sup>	$lm \cdot m^{-2} sr^{-1}$ i.e. $cd \cdot m^{-2}$ i.e. $nit$

FIG. A.5 – Tableau comparatif des quantités radiométriques et photométriques. Watt ( $W$ ). Lumen ( $lm$ ). Watt par mètre carré ( $W \cdot m^{-2}$ ). Candela ( $cd$ ). Watt par mètre carré par stéradian ( $W \cdot m^{-2} \cdot sr^{-1}$ ). Lumen par stéradian ( $lm \cdot sr^{-1}$ ).

### Correspondance entre les grandeurs radiométriques et photométriques

La figure A.5 présente les principales correspondances entre les grandeurs radiométriques et les grandeurs photométriques. Notons que, contrairement à la radiance, la luminance ne fait pas partie du Système International d'unité (SI) et que le candela est l'une des 7 quantités de base du SI<sup>6</sup>. Le lumen défini à partir du candela représente le flux lumineux émis par une source ponctuelle isotropique d'intensité lumineuse égale à 1 candela.

### Emission et réception de la lumière

La lumière émise sous forme de photons émane de sources lumineuses comme le soleil, les lampes ou encore des phénomènes naturels tels que le feu. Pour chaque source lumineuse, il est normalement nécessaire de connaître la distribution :

- spatiale,
- directionnelle,
- spectrale.

Un exemple simple est celui d'une source lumineuse ponctuelle placée en  $O_L$  qui émet donc dans toutes les directions avec une puissance spectrale  $\Phi_\lambda$ . L'irradiance reçue en un point  $x$  d'une surface est alors donnée par :

$$E_\lambda(x) = \frac{\Phi_\lambda \cos \theta}{4\pi r^2}$$

avec :

- $r$  distance entre  $x$  et la source lumineuse
- $\theta$  est l'angle entre la normale de la surface au point  $x$  et le vecteur  $\vec{xO}_L$ .

Intuitivement, on peut voir que le facteur  $\cos \theta$  est nécessaire pour tenir compte du fait qu'une surface qui fait face à la lumière recevra plus d'énergie lumineuse qu'une surface qui lui tourne le dos.

### Formulation implicite des longueurs d'onde. Espace de Couleur

On définit la radiance par la somme des radiances spectrales sur l'ensemble du domaine spectral :

$$L(x \rightarrow \omega) = \int_0^\infty L_\lambda(x \rightarrow \omega) d\lambda. \quad (\text{A.11})$$

<sup>6</sup>Les 7 quantités de base du SI sont :  $m$  (distance exprimée en mètre),  $kg$  (masse exprimée en kilogramme),  $s$  (temps exprimé en seconde),  $A$  (intensité électrique exprimée en ampère),  $K$  (température exprimée en degré Kelvin),  $mol$  (quantité de matière, nombre de moles) et  $cd$  (intensité lumineuse exprimée en candela).

Si l'on se place dans l'espace de couleur RGB<sup>7</sup>, la radiance est exprimée par une somme pondérée des 3 composantes  $R, G, B$  :

$$L(x \rightarrow \omega) = \alpha L_R(x \rightarrow \omega) + \beta L_G(x \rightarrow \omega) + \gamma L_B(x \rightarrow \omega) \quad (\text{A.12})$$

avec  $(\alpha, \beta, \gamma) \in \mathbb{R}^3$ .

$L_{R,G,B}$  peut prendre n'importe quelle valeur sur  $\mathbb{R}_+$ . Par conséquent, afin d'afficher la valeur de  $L(x \rightarrow \omega)$  sur un dispositif à plage limitée, il convient d'effectuer une transformation appropriée appelée *tone-mapping* [War94, Sch94c, TR93, DD00]. Lorsqu'il n'y a aucune ambiguïté permise, on omet la notation spectrale des quantités radiométriques.

## A.2 Définition quantique de l'énergie rayonnante spectrale

L'équation reliant la constante universelle de Planck et l'énergie d'un photon, quantité radiométrique élémentaire, permet d'établir la relation entre l'énergie du photon pour une longueur d'onde donnée :

$$e_\lambda = \frac{hc}{\lambda} \quad (\text{A.13})$$

avec

- $h$  la constante universelle de Planck ( $h \simeq 6,6261 \cdot 10^{-34} \text{ J.s}$ )
- $c$  la vitesse de la lumière dans le vide ( $c \simeq 299\,792\,458 \text{ ms}^{-1}$ ).

Il en découle deux quantités radiométriques :

- l'énergie rayonnante spectrale<sup>8</sup>  $Q_\lambda$ , exprimée en Joule par nanomètre ( $\text{J nm}^{-1}$ ), qui est définie pour un ensemble de  $n$  photons ayant la même longueur d'onde :

$$\begin{aligned} Q_\lambda &= n e_\lambda \\ Q_\lambda &= n \frac{hc}{\lambda}. \end{aligned} \quad (\text{A.14})$$

- l'énergie rayonnante<sup>9</sup>  $Q$ , exprimée en Joule ( $J$ ) qui est l'énergie obtenue en intégrant l'énergie rayonnante spectrale sur tout le spectre :

$$Q = \int_{\lambda=0}^{\infty} Q_\lambda d\lambda. \quad (\text{A.15})$$

## A.3 Définition du flux énergétique spectral en fonction de la température

Si la lumière est donnée par sa température, la formule de Planck rappelée ci-dessous permet de calculer son flux énergétique spectral par :

$$\Phi_\lambda = \frac{2\pi C_1}{\lambda^5 \left( e^{\frac{C_2}{\lambda T}} - 1 \right)} \quad (\text{A.16})$$

<sup>7</sup>RGB pour *Red, Green, Blue*. Le sigle français équivalent est RVB pour Rouge, Vert, Bleu.

<sup>8</sup>Le terme utilisé dans la littérature anglo-saxonne est *spectral radiant energy*.

<sup>9</sup>Le terme utilisé dans la littérature anglo-saxonne est *radiant energy*.

avec

- $T$  la température du corps donnée en degré Kelvin (K)
- $C_1 = h c^2$
- $C_2 = \frac{hc}{k}$
- $k$  la constante de Boltzmann ( $k = 1,380\,650\,5 \cdot 10^{-23} J/K$ ).





## Annexe B

# Une base alternative pour reconstruire l'irradiance dans l'espace des directions

Le but de cette annexe est de proposer une formulation mathématique rigoureuse d'une base pour reconstruire l'irradiance à partir de vecteurs d'irradiance. Cette base peut alors être utilisée comme alternative à l'interpolation proposée à la section 3.3.

### B.1 Grandeurs radiométriques vectorielles

Définissons les grandeurs vectorielles utilisées dans toute cette annexe.

On entend par radiance directionnelle au point  $x$  la quantité  $L_x(\omega)$  telle que :

$$L_x(\omega) = L(x \leftarrow \omega) \omega \tag{B.1}$$

où  $L(x \leftarrow \omega)$  représente la radiance incidente (un scalaire), repérée par la direction  $\omega$ , au point  $x$ .

En intégrant la radiance directionnelle sur un hémisphère  $\Omega_N$  orienté selon la normale  $N$  au point  $x$ , on définit le vecteur d'irradiance  $I_N(x)$  :

$$I_N(x) = \int_{\Omega_N} L_x(\omega) d\omega . \tag{B.2}$$

L'irradiance  $E(x)$ , quantité scalaire, au point  $x$  est liée au vecteur d'irradiance par la formulation suivante :

$$E(x) = I_N(x) \cdot N \tag{B.3}$$

où  $\cdot$  représente le produit scalaire.

Représenter efficacement la radiance incidente ou encore l'irradiance dans l'espace des directions (la sphère) est une problématique en rendu. Plus la représentation est précise, plus l'éclairage incident est correctement représenté. Une façon classique de faire est de projeter la radiance dans une base de fonctions définies sur la sphère telle que les harmoniques sphériques ou hémisphériques ou encore des ondelettes. Dans la suite du document nous proposons une nouvelle base dont l'élément radiométrique stocké est le vecteur d'irradiance.

## B.2 Base de fonctions sur la sphère

**Définitions** Soit  $\Phi$  un ensemble regroupant  $p$  fonctions définies sur la sphère :  $\Phi = \{\phi_0(\omega), \phi_1(\omega), \dots, \phi_{p-1}\}$ . Afin de pouvoir projeter une fonction dans  $\Phi$  il est nécessaire que  $\Phi$  soit une base. La projection est optimale et facilitée lorsque  $\Phi$  est une base orthonormalisée. Une manière de déterminer si  $\Phi$  est une base et son type (orthogonale, orthonormalisée, etc) est de calculer la matrice de corrélation (Gram Matrix)  $M_{ij}$  définie par :

$$M_{ij} = \int_S \phi_i(\omega) \phi_j(\omega) d\omega \quad (\text{B.4})$$

où  $S$  représente la sphère et donc tout l'espace des directions. En fonction de  $M$  on peut caractériser  $\Phi$  :

- si  $M$  est inversible alors  $\Phi$  est une base sinon  $\Phi$  est au mieux une famille génératrice
- si  $M$  est la matrice identité alors  $\Phi$  est une base orthonormée
- si  $M$  est diagonale (mais n'est pas la matrice identité) alors  $M$  est une base orthogonale
- sinon  $M$  est quelconque.

Lorsque  $M$  est quelconque mais inversible, une opération de projection dans la base est possible en utilisant une base duale. Une base duale consiste à définir un ensemble de fonctions  $\tilde{\phi}_k$  telles que :

$$\int_S \phi_i(\omega) \tilde{\phi}_k(\omega) d\omega = \begin{cases} 1 & \text{si } i = k \\ 0 & \text{sinon} \end{cases} \quad (\text{B.5})$$

Les  $\tilde{\phi}_k$  sont obtenues par combinaison linéaire des  $\phi_i$  :

$$\tilde{\phi}_k = \sum_j M_{kj}^{-1} \phi_j(\omega) \quad (\text{B.6})$$

En toute généralité, cette dernière équation que les coefficients  $\tilde{\phi}_k$  obtenus sont en fait une combinaison linéaire des coefficients  $\phi_i$ .

### Cas particuliers pour $\tilde{\phi}_k$

- si  $M$  est la matrice identité alors  $\forall k, \tilde{\phi}_k = \phi_k$ .
- si  $M$  est une matrice diagonale dont l'élément diagonal est  $\lambda_k$  alors  $\tilde{\phi}_k = \frac{1}{\lambda_k} \phi_k$ .

### Projection et Reconstruction

En toute généralité, projeter une fonction  $f(\omega)$  définie sur la sphère dans la base  $\Phi$  consiste à calculer  $p$  coefficients notés  $l_i$  et définis tels que :

$$l_i = \int_S f(\omega) \tilde{\phi}_i(\omega) d\omega. \quad (\text{B.7})$$

Ces coefficients permettent de reconstruire la fonction  $f$  à l'aide des fonctions de base :

$$f(\omega) \approx \sum_i^p l_i(\omega) \phi_i(\omega) \quad (\text{B.8})$$

### Application à la radiance vectorielle

Lorsque la fonction à projeter  $f(\omega)$  est la radiance vectorielle les coefficients  $l_i$  s'obtiennent de la même manière qu'à l'équation B.7 à la différence près que ces coefficients sont vectoriels :

$$l_i = \int_S \tilde{\phi}_i(\omega) L_x(\omega) d\omega \quad (\text{B.9})$$

et la reconstruction s'obtient de manière similaire à l'équation B.8.

### Reconstruction du vecteur d'irradiance

Afin de calculer le vecteur d'irradiance et donc l'irradiance en fonction des  $l_i$ , il est nécessaire de le définir dans l'espace des directions :

$$I_N(x) = \int_S \psi_N(\omega) L_x(\omega) d\omega. \quad (\text{B.10})$$

où  $\psi_N$  est un filtre hémisphérique défini tel que :

$$\psi_N(\omega) = \begin{cases} 1 & \text{si } \omega \cdot N > 0, \\ 0 & \text{sinon.} \end{cases} \quad (\text{B.11})$$

En utilisant l'équation B.2 avec l'équation B.8 on obtient :

$$\begin{aligned} I_N(x) &\approx \int_S \psi_N(\omega) \sum_i^p l_i(x) \phi_i(\omega) d\omega \\ &\approx \sum_i^p l_i(x) \underbrace{\int_S \psi_N(\omega) \phi_i(\omega) d\omega}_{\beta_i(N)}. \end{aligned}$$

On obtient donc une formulation simple du vecteur d'irradiance :

$$I_N(x) \approx \sum_i^p l_i(x) \beta_i(N) \quad (\text{B.12})$$

## B.3 Une base pour les vecteurs d'irradiance

Il est grand temps de donner une définition à  $\Phi$ . Soit  $\Phi$  la nouvelle base proposée définie comme suit :

$$\Phi = \begin{cases} \phi_0(\omega) = 1 \\ \phi_1(\omega) = \text{sign}(\omega \cdot [1, 0, 0]^T) \\ \phi_2(\omega) = \text{sign}(\omega \cdot [0, 1, 0]^T) \\ \phi_3(\omega) = \text{sign}(\omega \cdot [0, 0, 1]^T) \end{cases} \quad (\text{B.13})$$

avec  $\text{sign}(x) : \mathbb{R} \rightarrow \{-1, 1\}$ , la fonction définie par :

$$\text{sign}(x) = \begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{sinon} \end{cases}$$

### Orthogonalité et Normalisation

Montrons que  $\forall i, j$  avec  $i \neq j$  :  $\int_S \phi_i(\omega) \phi_j(\omega) d\omega = 0$ .

Posons  $i = 0$ , alors  $\forall j \in \llbracket 1, 3 \rrbracket$ , on a :

$$\int_S \phi_0(\omega) \phi_j(\omega) d\omega = \int_S \phi_j(\omega) d\omega = \int_{+\Omega_j} \phi_j(\omega) d\omega + \int_{-\Omega_j} \phi_j(\omega) d\omega = 0 \quad (\text{B.14})$$

avec  $+\Omega_j$  (resp.  $-\Omega_j$ ) représentant l'hémisphère orienté positivement (resp. négativement) selon la fonction  $\phi_j$ . En décomposant de manière similaire le domaine d'intégration (la sphère) en quartier, on peut démontrer que :

$$\forall i, j \in \llbracket 1, 3 \rrbracket \times \llbracket 1, 3 \rrbracket \text{ avec } i \neq j : \int_S \phi_i(\omega) \phi_j(\omega) d\omega = 0 \quad (\text{B.15})$$

Par conséquent,  $\Phi$  constitue une base orthogonale. En revanche  $\Phi$  n'est pas orthonormalisée puisque :

$$\forall i, \int_S \phi_i(\omega) \phi_i(\omega) d\omega = \int_S d\omega = 4\pi \neq 1. \quad (\text{B.16})$$

**Matrice de corrélation** Compte tenu des équations précédentes, la matrice de corrélation  $M_{ij}$  pour notre base est simplement :

$$M_{ij} = \begin{pmatrix} 4\pi & 0 & 0 & 0 \\ 0 & 4\pi & 0 & 0 \\ 0 & 0 & 4\pi & 0 \\ 0 & 0 & 0 & 4\pi \end{pmatrix}$$

Par conséquent les fonctions de base duales  $\tilde{\phi}_k$  s'expriment par :  $\tilde{\phi}_i = \frac{1}{4\pi} \phi_i$ .

### Projection et reconstruction

Dans cette base le calcul des coefficients  $l_i(x)$  est équivalent à :

$$\begin{aligned} l_i(x) &= \int_S L_x(\omega) \tilde{\phi}_i(\omega) d\omega \\ &= \frac{1}{4\pi} \int_S L_x(\omega) \phi_i(\omega) d\omega. \end{aligned}$$

Les coefficients  $\beta_i(N)$  peuvent être évalués numériquement mais pour notre base une formulation analytique existe :

$$\forall i \in \llbracket 1, 3 \rrbracket \beta_i(N) = 2\pi \left( 1 - \frac{2 * \arccos(N \cdot \omega_i)}{\pi} \right) \quad (\text{B.17})$$

où  $\omega_i = [1, 0, 0], [0, 1, 0], [0, 0, 1]$ . Remarque :  $\beta_0(N) = 2\pi$ .

### Intéret Immédiat

La nouvelle base proposée a donc l'avantage d'être orthogonale. De plus, la projection des vecteurs d'irradiance reste simple et sûrement moins coûteuse (à vérifier) qu'une projection dans la base des harmoniques sphériques. En terme de mémoire, le nombre de vecteurs d'irradiance par sommet est réduit bien qu'il faille désormais stocker une table supplémentaire pour représenter les  $\beta_i(N)$ . Pour le rendu sur carte graphique cette table peut être stockée dans une seule texture.

## Intégration de la BRDF

On peut calculer une autre forme pour les coefficients  $\beta_i(N)$  qui intègrè la BRDF sous certaine condition.

Considérons la  $R(x \rightarrow \omega_o)$ , la radiance directionnelle au point  $x$  dans la direction  $\omega_o$  :

$$R(x \rightarrow \omega_o) = \int_S f(\omega, \omega_o) L(x \leftarrow \omega) \omega \psi_N(\omega) d\omega \quad (\text{B.18})$$

où  $f$  représente la BRDF au point  $x$ . En utilisant la base  $\Phi$ , on peut exprimer  $R(x \rightarrow \omega_o)$  en fonction des coefficients  $l_i(x)$  :

$$R(x \rightarrow \omega_o) = \sum_i l_i(x) \underbrace{\int_S f(\omega, \omega_o) \phi_i(\omega) \psi_n(\omega) d\omega}_{\gamma_i(N, \omega_o)} \quad (\text{B.19})$$

La fonction  $\gamma_i(N, \omega_o) : \mathbb{R}^4 \rightarrow \mathbb{R}$  est complexe et difficilement exploitable. En revanche, elle est simplifiable pour certain type ou représentation de BRDF.

Dans le cas diffus,  $f(\omega, \omega_o) = \frac{k_d}{\pi}$ . On obtient alors immédiatement  $\beta_i(N) = \int_S \phi_i(\omega) \psi_N(\omega) d\omega$  et par conséquent  $R(x \rightarrow \omega_o)$  devient égal à :

$$R(x \rightarrow \omega_o) = \frac{k_d}{\pi} \sum_i l_i(x) \beta_i(N) .$$

A l'instar de ce qui est fait pour les PRT (e.g. [WTL04]) une BRDF quelconque peut se décomposer en produit de deux fonctions 2D :

$$f(\omega, \omega_o) = \sum_k^K g_k(\omega_o) h_k(\omega)$$

où  $K$  est le nombre de terme utilisé pour décomposer la BRDF. Naturellement plus  $K$  est élevé plus l'approximation est précise. Dans le domaine des PRT,  $K$  est en général égal à 4 maximum pour des BRDFs dites glossy. Cette décomposition permet alors de simplifier la fonction  $\gamma$  pour retrouver un produit de deux fonctions 2D et exprimer  $R(x \rightarrow \omega_o)$  :

$$R(x \rightarrow \omega_o) = \sum_i l_i(x) \sum_k g_k(\omega_o) \underbrace{\int_S h_k(\omega) \phi_i(\omega) \psi_N(\omega) d\omega}_{\beta_i(N)} .$$

## B.4 Retour sur la proposition de la section 3.3.3

Rappelons qu'aux section 3.3 et 3.3.3 nous proposons de stocker, pour chaque échantillon de la structure de cache, 6 vecteurs d'irradiance notés  $I_{\pm X}$  (resp.  $I_{\pm Y}$  et  $I_{\pm Z}$ ) où  $(\pm X, \pm Y, \pm Z)$  représentent les six directions principales du repère cartésien. Pour un point  $x$  et sa normale  $N = (n_x, n_y, n_z)$ , on peut interpoler le vecteur d'irradiance  $I_N(x)$  à l'aide des 6 vecteurs de la manière suivante :

$$I_N(x) \approx I_{\pm X}(x) n_x^2 + I_{\pm Y}(y) n_y^2 + I_{\pm Z}(x) n_z^2 . \quad (\text{B.20})$$

Le choix du signe de  $X$  (resp. du signe de  $Y$  et de  $Z$ .) se fait en fonction du signe de  $n_x$  (resp.  $n_y$  et  $n_z$ ). Dans ce cas, la base  $\Phi$  est composée de 6 fonctions de bases :  $\Phi = \psi_{+X}, \psi_{-X}, \psi_{+Y}, \psi_{-Y}, \psi_{+Z}, \psi_{-Z}$ .

Le problème de ces fonctions de bases est qu'elles ne forment pas une base. Ceci peut notamment se démontrer en remarquant que :  $\phi_{+X} + \phi_{-X} = \phi_{+Y} + \phi_{-Y} = \phi_{+Z} + \phi_{-Z} = 1$  Il y a donc une dépendance linéaire entre éléments de  $\Phi$  qui ne peut donc pas être une base.

## Annexe C

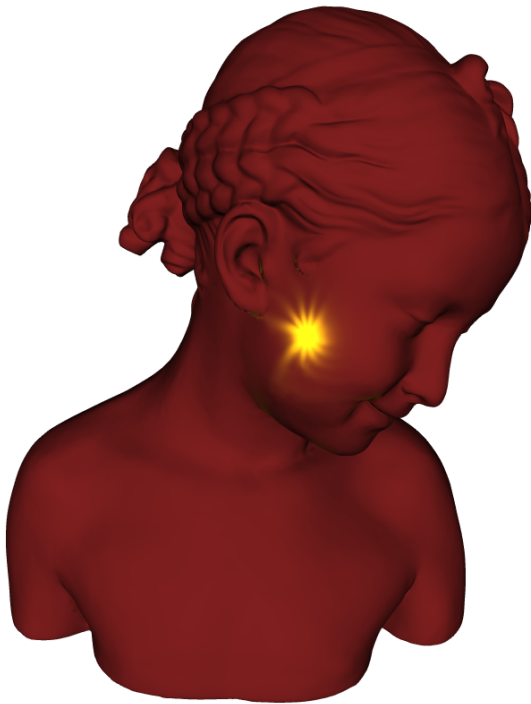
# Article de travail sur l'édition vectorielle de BRDF

# Interactive Lobe Design for Visually Plausible BRDFs

Submission Number 217

## Abstract

*This paper introduces a new approach for the generation of visually plausible BRDFs. This approach is based on an interactive creation and edition of BRDF lobes, mostly responsible for the appearance of surface highlights. The lobe definition is built on a new parameterization, based on the mirror direction of the incoming light. Through a set of simple and intuitive parameters, the user can modify the lobes' behavior. The effectiveness of our solution is illustrated with various examples.*



**Figure 1.** A complex highlight designed with our system.

## 1. Motivation

In computer graphics, the Bidirectional Reflectance Distribution Function (BRDF) plays a central role for the visual appearance of 3D surfaces. A large amount of research

has been devoted to develop analytical BRDF models or to numerically estimate the BRDF from measured data of real world materials. As pointed out by Lewis [13], *physically plausible* BRDFs must satisfy two constraints: (1) reciprocity (*i.e.*, inverting light and viewing directions does not change the reflected radiance), and (2) energy conservation (*i.e.*, the total reflected energy is lower than or equal to the incoming energy). By relaxing these two constraints, a broader class of *visual plausible* BRDFs can be obtained.

Early empirical reflection models do not satisfy the energy conservation constraint [18, 3], but are still very popular for real-time rendering as they provide a plausible appearance. Energy conservation is mostly required in global illumination algorithms in order to guarantee convergence. This has led researchers either to derive theoretically sound models from the laws of optics [5, 8, 1], or to improve the earlier models [24, 21, 12]. Lafortune and Willems [11] and Neumann *et al.* [14] have also shown that energy conservation can be introduced into a model that is only reciprocal.

The main goal of this paper is to provide a simple, yet flexible BRDF model that can be easily created or edited by a user. Because the physically plausible constraints can limit the freedom in designing the reflection appearance of 3D surfaces, we choose to neglect both the energy conservation and reciprocity constraints.

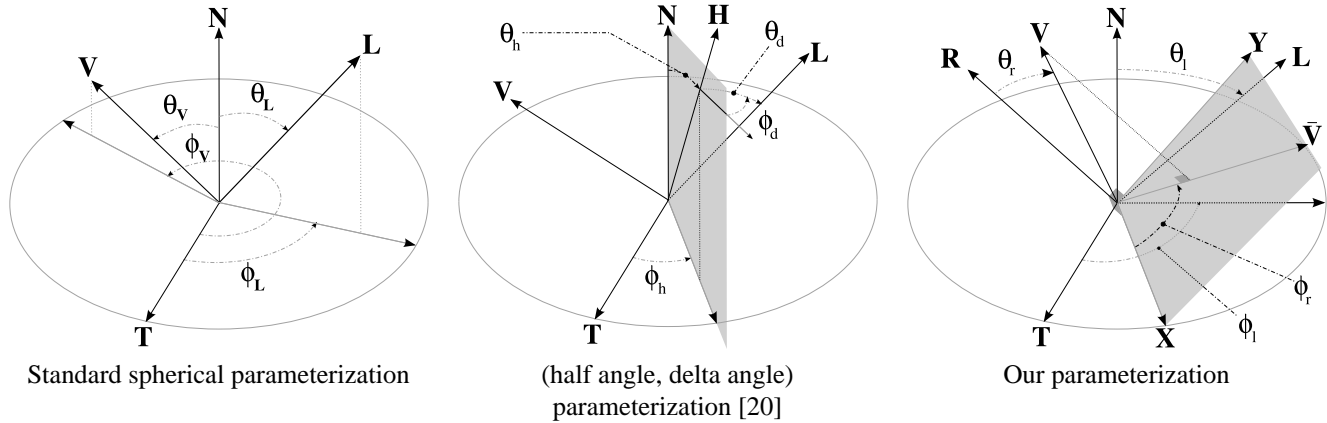
Much of the surface appearance due to a BRDF is captured by its highlights. These highlights, usually generated by specular or glossy reflection, frequently fit simple shapes (*e.g.*, a lobe) oriented according to the incoming direction of light. Our approach for BRDF design is based on the following assumptions. First, the design is performed relatively to a given lighting direction. Second, the user directly designs each lobe shape and color, and therefore, controls the resulting highlight.

Based on these assumptions, this paper introduces three main contributions:

1. A new parameterization of the BRDF, based on the mirror direction of the incoming light.
2. A new analytical description of a lobe, suitable for a large range of shapes.
3. An interactive BRDF creation and edition process.

This paper is organized as follows. First, we briefly in-





**Figure 2.** Configuration of different BRDF parameterizations.

roduce previous work on BRDF edition and parameterization. Then, we present the new parameterization, the new analytical lobe shape, followed by our interactive edition tool. Finally, we illustrate our approach by presenting some results, and discuss the pros and cons of our BRDF editing system.

## 2. Previous Work

### BRDF Edition

If a large body of work has been published on BRDF models [18, 3, 5, 8, 1, 24, 21] and BRDF fitting [12, 15], much less work has been devoted to BRDF editing.

Early work on BRDF editing done by Poulin and Fournier [19] has been recently extended by Colbert *et al.* [4]. In their tool, called *BRDF-Shop*, different “painting” operations are introduced for the design of highlights. The resulting BRDF corresponds to the fit of a sum of Ward lobes [24]. Complex shapes can thus require a large number of lobes. Using a parameterization based on the half vector [3], Edwards *et al.* [7], extending the work of Ashikhmin *et al.* [1], allow the creation of a BRDF by designing a probability distribution function of normals. The resulting BRDF is energy conserving, not reciprocal, but still visually plausible. Furthermore, the shape of the lobe (*i.e.*, the highlight) is difficult to control as it is indirectly edited. A similar indirect control has also been proposed by Neumann *et al.* [14], where the shape of the lobe is defined on the tangent plane of the surface. This process is rather unintuitive as the projection on the tangent plane involves strong deformations of the lobe.

In another recent work related to BRDF editing, Ben-Artzi *et al.* [2] project existing BRDFs into a given basis and factorize them in a set of 1D curves that can be directly edited, even under complex illumination. Ngan *et al.* [16] present an intuitive interface for the naviga-

tion through different BRDFs and their parameters. Pelacini and Lawrence [17] develop an interactive tool for local edition of spatially varying BRDFs. None of these approaches offer the possibility of a direct choice of the highlight shapes.

The most closely related work to our goal comes from non-photorealistic rendering. By neglecting some of the constraints due to realism, new shading approaches have been introduced in order to increase user freedom. With the *Lit Sphere* [23], Sloan *et al.* introduce a new way for defining the appearance of a surface. The solution painted on a sphere is used as a texture, reprojected on 3D surfaces, taking into account the similarity of the configuration between the viewpoint and the normal. Therefore this approach is only possible for view-dependent lighting. Anjyo *et al.* [9] present a new solution for tweaking the shape of a highlight reflection in a cartoon-like shading. However they do not really edit the BRDF: they locally move, scale, and split the light source in order to obtain the requested shape.

### BRDF Parameterization

The standard spherical parameterization (see Figure 2-left), based on the light and view directions (*e.g.*, [10]), suffers from undefined parameters (the azimuthal angles) in two configurations, when the light or view direction is aligned with the normal.

Most of the BRDF parameterizations are based on the half vector, introduced by Blinn [3], in order to create more realistic reflection models. This half vector (*cf.* Figure 2-middle) has proved efficient for fitting [15] and accuracy. Unfortunately, this approach [20] still suffers from undefined parameters (two of the four required angles) in two configurations: when the half vector is equal to the normal, or when the view and light directions are equal. Furthermore, it is not well suited for the definition of a lobe shape for each light direction, since the half vector is both view-

dependent and light-dependent.

For direct control on highlights, and thus of the corresponding lobes, it seems more suited to use the classical parameterization based on the mirror-reflected vector [18]. In this paper, we will show that based on this approach, we can provide a parameterization (see Figure 2-right) with only one un-determined parameter for a single configuration: when the light source is aligned with the normal.

### 3 A New BRDF Parameterization

In this paper, we will use the following notation:  $\mathbf{N}$  the normal,  $\mathbf{L}$  the lighting vector pointing toward the light source,  $\mathbf{R}$  the direction of mirror-reflection for  $\mathbf{L}$ , and  $\mathbf{V}$  the viewing direction, as illustrated in Figure 2-right. In all equations, each vector noted in boldface is normalized to unit length, “ $\cdot$ ” denotes a dot product, and “ $\times$ ” a cross product.

As in the standard spherical parameterization, the light direction is controlled by two angles,  $\theta_l$  and  $\phi_l$ , corresponding respectively to the azimuthal angle measured from tangent vector  $\mathbf{T}$ , and the zenithal angle  $\theta_l$  measured from normal vector  $\mathbf{N}$ . They are defined as

$$\begin{cases} \cos \theta_l = \mathbf{L} \cdot \mathbf{N} \\ \cos \phi_l = \mathbf{L} \cdot \mathbf{T} \\ \sin \phi_l = \mathbf{L} \cdot \mathbf{B} \end{cases}$$

where  $\mathbf{B} = \mathbf{N} \times \mathbf{T}$ .

For the parameterization of the specular lobe, we introduce two angles  $\theta_r$  and  $\phi_r$ , respectively representing the deviation of the viewing direction  $\mathbf{V}$  from the reflected direction  $\mathbf{R}$ , and the orientation of this deviation. They are defined as:

$$\begin{cases} \cos \theta_r = \mathbf{V} \cdot \mathbf{R} \\ \cos \phi_r = \mathbf{V} \cdot \mathbf{X} \\ \sin \phi_r = \mathbf{V} \cdot \mathbf{Y} \end{cases}$$

where  $\mathbf{X}$  and  $\mathbf{Y}$  are defined as

$$\begin{cases} \mathbf{Y} = \frac{\mathbf{N} - (\mathbf{N} \cdot \mathbf{R}) \mathbf{R}}{\sqrt{1 - (\mathbf{N} \cdot \mathbf{L})^2}} \\ \mathbf{X} = \mathbf{Y} \times \mathbf{R} \end{cases} .$$

Intuitively,  $\theta_r$  and  $\phi_r$  parameterize a spherical cap where  $\mathbf{R}$  is the North pole axis. This is somehow related to the (half angles, delta angles) parameterization proposed by Rusinkiewicz [20]. As will be detailed in the next section,  $\theta_r$  and  $\phi_r$  may naturally be used as input parameters for a polar representation.

As  $\mathbf{L} \cdot \mathbf{N}$  and  $\mathbf{V} \cdot \mathbf{R}$  can always be computed,  $\theta_l$  and  $\theta_r$  are always defined. On the other hand, this parameterization presents a discontinuity at normal incidence, when

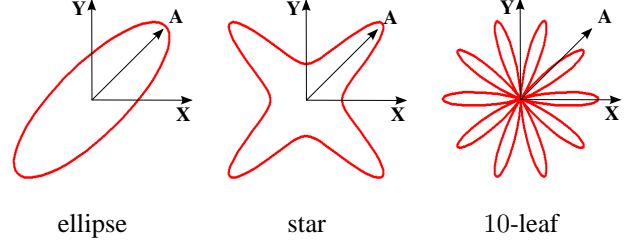


Figure 3. Different bases for the lobe cone.

$\mathbf{N} \cdot \mathbf{L} = 1$ : since  $\mathbf{L} \cdot \mathbf{T} = \mathbf{L} \cdot \mathbf{B} = 0$ ,  $\phi_l$  is undefined, and since  $\mathbf{Y}$  cannot be computed,  $\phi_r$  is also undefined. Consequently, in this limit case, there is no orientation which means the shape of the lobe has to be independent of  $\phi_l$  and  $\phi_r$ . Thus, we have to force the lobe to be isotropic for this particular configuration (only the lobe, but not the BRDF). While this seems to restrict the design freedom, it is a coherent behavior of classical analytical BRDF models where anisotropy vanishes at normal incidence.

### 4 Analytical Lobe Shape and Color

One of our assumptions is that the lobe shape is designed for a given lighting direction. We have also shown in the previous section that the shape has to be isotropic at normal incidence. Thus, each freely defined shape has to morph to an isotropic function in the limit case in order to remove the dependency on the orientation angles  $\theta_r$  and  $\theta_l$ . In polar coordinates, an isotropic function only depends on the radius, and not on the angle. The morphing process has therefore only to ensure that the dependency on the angle is removed in the limit case.

The two parameters  $\theta_r$  and  $\phi_r$  are used to describe the reflection lobe.  $\theta_r$  can be easily converted to a value similar to a polar radius  $\rho = 1 - \cos \theta_r$ , and we keep  $\phi_r$  as the polar angle.

#### 4.1 Lobe Shape

The lobe is represented as a cone with a base controlled by a polar curve  $r(\phi_r)$ . To be consistent with classical BRDF models, the intensity  $I(\rho, \phi_r)$  of the lobe should monotonously decrease for a maximum value at the cone apex (when  $\rho = 0$ ) to 0 at the cone base (when  $\rho = r(\phi_r)$ ). This can be easily obtained by defining

$$I(\rho, \phi_r) = w(t) \quad \text{with} \quad t = \rho/r(\phi_r)$$

and  $w(t)$  is any monotonously decreasing function in the range  $[0, 1]$  without loss of generality. One classical choice (like in Phong’s model) is to use the clamped power function  $w(t) = (1 - t)^n$ ,  $t \in [0, 1]$ . In this paper, we use the

more efficient rational function proposed by Schlick [22]:

$$w(t) = \begin{cases} 0, & t > 1 \\ 1 - \frac{t}{n - nt + t}, & t \in [0, 1] \end{cases} \quad (1)$$

In both equations,  $n$  is similar and controls the shininess. Note also that more complex intensity variations can be obtained with arbitrary functions  $w(t)$ .

### Interpolation for limit case

For the limit case, lighting is incident to the normal ( $\theta_l = 0$ ),  $r(\phi_r)$  has to be constant (i.e.,  $r(\phi_r) = r$ ), leading to a circular base of radius  $r$ .

In order to morph the generic base to a circular one when  $\theta_l = 0$ , a simple interpolation can be done:

$$r(\phi_r, \theta_l) = r(\phi_r) + (r - r(\phi_r)) f(\theta_l),$$

where  $f(\theta_l)$  is controlling the interpolation between the curve defined for the user chosen lighting direction  $\theta_l = \theta_0$  and the circular one of radius  $r$  at normal incidence. This function can be arbitrarily chosen by the user, as long as  $f(0) = 1$  and  $f(\theta_0) = 0$  to get the interpolation of the extremal cases. In this paper, we use:

$$f(\theta_l) = \begin{cases} \frac{\cos \theta_l - \cos \theta_0}{1 - \cos \theta_0}, & \theta_l \leq \theta_0 \\ 0, & \theta_l > \theta_0 \end{cases}$$

Unfortunately, like in all morphing processes, such a simple interpolation does not provide a visually pleasing transition from any generic curve to a circle. Therefore, the morphing has to be more specifically defined for each family of curves.

### Polynomial Star-shaped Curves

An easy way to obtain visually pleasing morphing between an arbitrary shape and a circle is to define a parametric family of shape curves  $r(\phi_r)$  able to generate a circle for a specific configuration of the parameters. So by correctly interpolating the parameters, a circular lobe can be obtained for normal incidence.

With our lobe parameterization, the shape curve is defined as a polar function  $r(\phi_r)$ , so to get a circle, we simply have to define a constant function  $r(\phi_r) = r$ . This means that any family of curves containing a constant function for a given set of its degrees of freedom, can be used as a basis for our shape curves. In this section, we present some analytical shape curves that we have currently implemented in our BRDF design tool. However, as mentioned above, almost any parametric family of polar functions can be used. One can even imagine a free-form modeling of the shape curve, using for instance the *spline-based polar curves* proposed by Crespin et al. [6].

The following polynomial Star-shaped functions are used:

$$\text{Ellipse: } \frac{1}{r_{a,b}^2(\phi_r)} = \frac{u(\phi_r)^2}{a^2} + \frac{1-u(\phi_r)^2}{b^2}$$

$$\text{Star: } \frac{1}{r_{a,b}(\phi_r)} = \frac{1}{a^2} + \left(\frac{1}{a^2} - \frac{1}{b^2}\right) 4u(\phi_r)^2(1 - u(\phi_r)^2)$$

In these functions,  $u(\phi_r)$  corresponds to a rotation of the cone base. It is defined as:

$$u(\phi_r) = \bar{\mathbf{V}} \cdot \mathbf{A}$$

where  $\bar{\mathbf{V}} = (\cos \phi_r, \sin \phi_r)$  (see Figure 2-right), and  $\mathbf{A}$  is the main axis of the curve. For an ellipse,  $\mathbf{A}$  corresponds to the direction of the major axis.

In order to smoothly blend from these functions to a circular base, we use the following interpolation on parameters  $a$  and  $b$ :

$$\begin{cases} a(\theta_l) &= a + (r - a) f(\theta_l) \\ b(\theta_l) &= b + (r - b) f(\theta_l) \end{cases}$$

### $k$ -leaf Rose Curves

We also use the well-known  $k$ -leaf rose curves:

$$r_{a,b}(\phi_r) = a + b * \cos(k \arccos(u_{\phi_r})),$$

where  $k$  defines the number of “leaves”, and with the following interpolation scheme:

$$\begin{cases} a(\theta_l) &= a \\ b(\theta_l) &= (1 - f(\theta_l)) b \end{cases}$$

which morphs the  $k$ -leaf rose curve to a circle.

## 4.2 Color Variation

Similarly to the shape variation, we can easily introduce color variations in the highlights. The user simply defines a color for different lighting directions, and interpolates in-between them. In our model, we define a color  $\mathbf{c}_0$  for a light direction  $\theta_l = \theta_0$  and  $\mathbf{c}$  for normal incidence. The resulting color of the lobe is:

$$\mathbf{c}(\theta_l) = \mathbf{c}_0 + (\mathbf{c} - \mathbf{c}_0) f(\theta_l).$$

## 4.3 Retro-reflection

In this model, we can easily add a new lobe to define retro-reflection effects. This new lobe has the same parameterization than the reflection lobe, except that it is centered around  $\mathbf{L}$  ( $\mathbf{L}$  replaces  $\mathbf{R}$  in Equations 3).

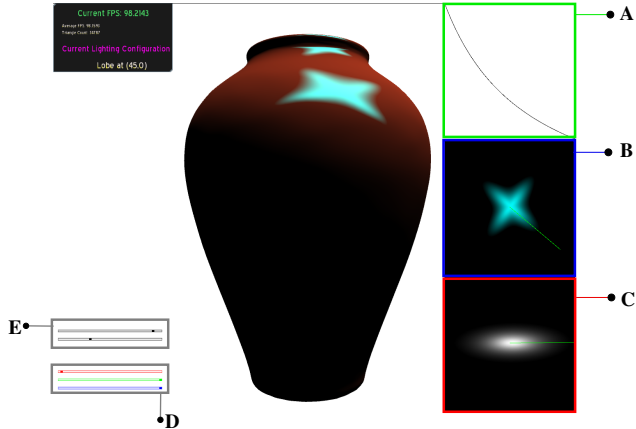


Figure 4. General view of our software.

## 5 Interface for User Edition

Once the user has selected a family of curves for the lobe shape (see Section 4.1), he can easily and interactively edit its different parameters. In this section, we briefly present the interactive tool (see Figure 4) that we have developed for the lobe edition. On the left, the main window shows the rendered surface, and below, the sliders (area **D**) control the color of the lobe and the sliders (area **E**) control the shape of the lobe through parameters  $a$  and  $b$  introduced in Section 4.1. In the three areas on the right, the user can control the orientation of the lobes (areas **B** and **C**) and their shininess (area **A**).

Area **A** presents the curve  $w(t)$  controlling the shininess of the BRDF. Parameter  $n$  is modified through edition of this curve.

In area **B**, the user can rotate the main axis of the lobe shape for the current light direction (except for the normal incidence). This window also shows the resulting combination of the lobe shape and the shininess function. In area **C** the user can similarly control the shape rotation of the retro-reflection lobe.

In order to edit shininess and color between the current light direction and the incident direction, the user can switch between these two configurations.

## 6 Results and discussion

In this section, we present and discuss some results obtained with our lobe design tool. All editing operations (color in Figure 5, orientation in Figure 7, shininess in Figure 6) are interactive<sup>1</sup> (from 19 to 60 frames per second, depending on the 3D surface complexity), giving immediate feedback to the user.

<sup>1</sup>On a Pentium4 3.0 GHz using a GeForce Quadro 3450 FX.

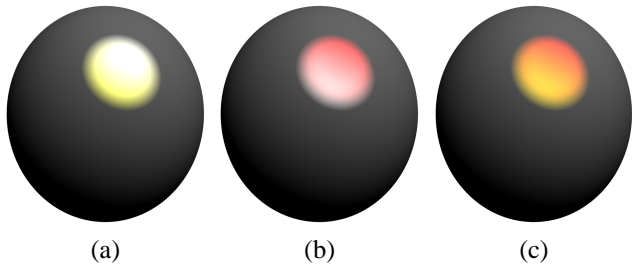


Figure 5. (a) Lobe color set to yellow for  $\theta_l = \frac{\pi}{4}$ . (b) Lobe color set to red for  $\theta_l = 0$ . (c) Final result when the two configuration colors are changed.

We have experimented with three lobe shape curves: an ellipse in Figures 6, 7, and 8; a star shape in Figure 9(a); and a 10-leaf curve in Figures 1 and 9(d). Creating such lobe shapes with an inverse fitting process, as done in *BRDF-Shop* [4], becomes quickly impracticable because of the large number of lobes to be fit (one for each leaf or one for each star's branch). The distortions of the leaves in Figure 1 are due to the quick variation of the normals around the center of the highlight, as well as the very thin lobe present on the upper edge of the vase in Figure 9.

As shown in Figures 7 and 8, where the lobe shape is edited and orientated, our system allows the simulation of highlights similar to anisotropic BRDFs without an underlying anisotropic BRDF. Note that for such a fake anisotropic effect, the tangent vector is not required on the surface, which makes this effect particularly interesting for implicit surfaces, where it can be tedious to define consistent tangent vectors all over the surface.

The shininess edition (Figure 6) consists in a simple edition of a corresponding curve (as defined in Equation 1) that controls the speed of the intensity variation, from 1 at the center of the lobe to the 0 on its border. With the simple ellipse curve and the shininess control, it is easy, as shown in Figure 5, to manually adjust (by setting the values of  $a$  and  $b$  to be equal) the parameters of our system to obtain a lobe very similar to Phong.

Finally, to allow edition of color variations in Figure 5, the user chooses separate colors for the  $\theta_l = 0$  and  $\theta_l = \pi/4$  configurations. This results in varying colored lobe suggesting an approximate appearance of a dispersion effect due to thin layers.

## 7 Conclusion and Future Work

In this paper, we have introduced a new parameterization and a new analytical BRDF model. The parameterization is based on the mirror-reflection of the light direction. Around this direction, the lobe is controlled by a function in

polar coordinates that can be selected by the user for a given light direction. We ensure the reciprocity by morphing the general lobe shape to a circular one for normal incidence lighting.

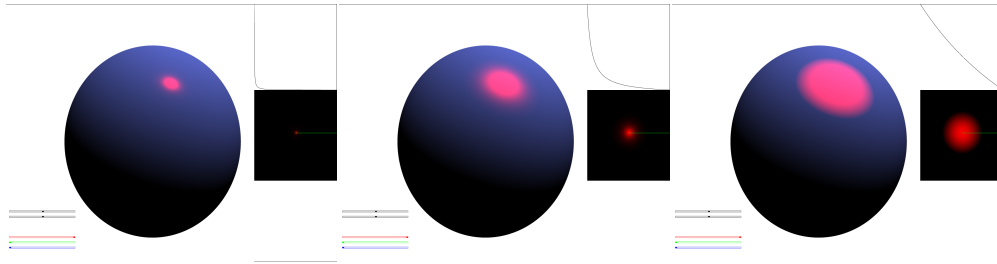
By selecting a polar function for the lobe description, the user can create a wide range of highlight shapes. Color, shininess, orientation, and parameters of the lobe shape can be interactively edited in order to reach the desired effect, as illustrated by the presented results.

In future work, we would like to investigate how dispersion or similar color effects could be simulated by allowing the lobe color to depend on lobe angles  $\theta_r$  and  $\phi_r$ . Second, we want to create anisotropic BRDFs, with fully free-form lobe shapes. In this purpose, the interpolation scheme has to be also improved.

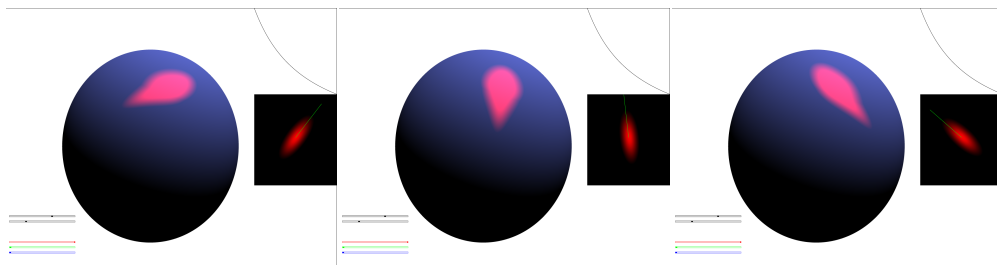
Finally, we believe that we can also create energy-conserving BRDFs. The main difficulty to achieve this goal, is to compute a normalization factor (like in [11] or [14]).

## References

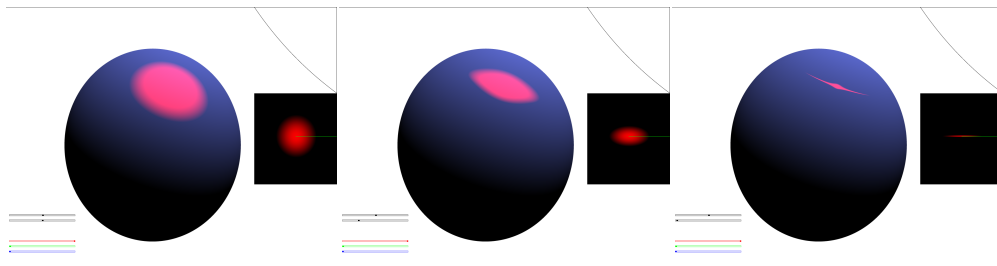
- [1] M. Ashikhmin, S. Premoze, and P. Shirley. A microfacet-based BRDF generator. In *ACM SIGGRAPH '00*, pages 65–74, 2000.
- [2] A. Ben-Artzi, R. Overbeck, and R. Ramamoorthi. Real-time BRDF editing in complex lighting. *ACM Trans. Graph.*, 25(3):945–954, 2006.
- [3] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH '77*, pages 192–198, 1977.
- [4] M. Colbert, S. Pattanaik, and J. Krivánek. BRDF-Shop: Creating Physically Correct Bidirectional Reflectance Distribution Functions. *IEEE Computer Graphics and Applications*, 26(1):30–36, 2006.
- [5] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, 1982.
- [6] B. Crespín, C. Blanc, and C. Schlick. Implicit Sweep Objects. *Computer Graphics Forum*, 15(3):165–174, 1996.
- [7] D. Edwards, S. Boulos, J. Johnson, P. Shirley, M. Ashikhmin, M. Stark, and C. Wyman. The halfway vector disk for BRDF modeling. *ACM Trans. Graph.*, 25(1):1–18, 2006.
- [8] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. A comprehensive physical model for light reflection. In *ACM SIGGRAPH '91*, pages 175–186, 1991.
- [9] K. ichi Anjyo, S. Wemler, and W. Baxter. Tweakable light and shade for cartoon animation. In *NPAR '06: Proc. international symposium on Non-photorealistic animation and rendering*, pages 133–139. ACM Press, 2006.
- [10] J. Kautz and M. D. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Proc. Eurographics Workshop on Rendering*, pages 247–260. Springer, June 1999.
- [11] E. P. Lafortune and Y. D. Willems. Using the Modified Phong BRDF for Physically Based Rendering. Technical Report CW197, Katholieke Universiteit Leuven, Belgium, 1994.
- [12] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *ACM SIGGRAPH '97*, pages 117–126, 1997.
- [13] R. R. Lewis. Making Shaders More Physically Plausible. *Computer Graphics Forum*, 13(2):109–120, 1994.
- [14] L. Neumann, A. Neumann, and L. Szirmay-Kalos. Reflectance Models with Fast Importance Sampling. *Computer Graphics Forum*, 18(4):249–265, 1999.
- [15] A. Ngan, F. Durand, and W. Matusik. Experimental Analysis of BRDF Models. In *Proc. Eurographics Symposium on Rendering*, pages 117–226, 2005.
- [16] A. Ngan, F. Durand, and W. Matusik. Image-driven Navigation of Analytical BRDF Models. In *Proc. Eurographics Workshop on Rendering*, pages 399–408, June 2006.
- [17] F. Pellacini and J. Lawrence. AppWand: Editing Measured Materials using Appearance-Driven Optimization. *ACM Trans. Graph.*, 26(3), 2007.
- [18] B. T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975.
- [19] P. Poulin and A. Fournier. Painting Surface Characteristics. In *Proc. Eurographics Workshop on Rendering*, pages 160–169. Springer, June 1995.
- [20] S. Rusinkiewicz. A New Change of Variables for Efficient BRDF Representation. In *Proc. Eurographics Rendering Workshop*, pages 11–22. Springer, 1998.
- [21] C. Schlick. An Inexpensive BRDF Model for Physically-Based Rendering. *Computer Graphics Forum*, 13(3):233–246, 1994.
- [22] C. Schlick. *Graphics Gems V*, chapter A Fast Alternative to Phong's Specular Model, pages 385–384. Morgan Kaufman, 1994.
- [23] P.-P. Sloan, W. Martin, A. Gooch, and B. Gooch. The Lit Sphere: A Model for Capturing NPR Shading from Art. In *Proc. Graphics Interface*, pages 143–150, 2001.
- [24] G. J. Ward. Measuring and modeling anisotropic reflection. In *ACM SIGGRAPH '92*, pages 265–272, 1992.



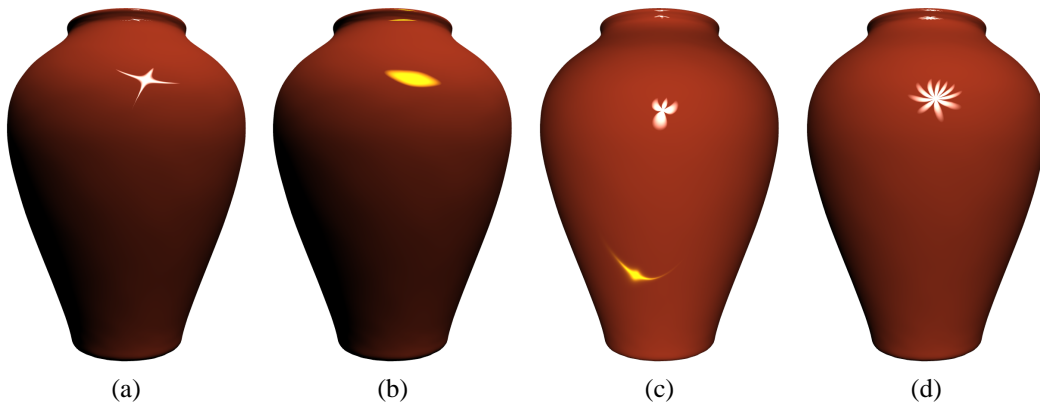
**Figure 6.** Lobe shininess editing.



**Figure 7.** Lobe orientation editing.



**Figure 8.** Lobe shape editing.



**Figure 9.** (a) Highlights designed with the star curve, (b) the ellipse, (c) 4-leaf, and (d) 10-leaf.

## Annexe D

# Article de travail sur la structure de cache

# Volumetric and Vectorial Representation for Indirect Illumination

paper\_182

---

## Abstract

*This paper introduces a volumetric representation to approximate indirect illumination. A 3D grid of irradiance vectors is first constructed via particle tracing. The indirect illumination within a voxel is interpolated from its associated irradiance vectors.*

*This volumetric and vectorial representation is more independent of the local variations of geometric and photometric properties of a scene, and captures low-varying diffusely inter-reflected lighting. This properties are therefore well suited for higher-quality caching schemes compared to state of the art solutions as demonstrated by our results. Furthermore, thanks to its low-cost evaluation and compactness, this structure can be easily adapted to current programmable hardware for real-time display of indirect illumination.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and RealismShading; Radiosity; RayTracing; I.3.3 [Computer Graphics]: Picture/Image Generation

---

## 1. Introduction

Among all global illumination algorithms, *Photon Mapping* [Jen01] has become very popular, thanks to its optimized structures and its ability to deal in an integrated manner with such varied complex lighting effects as diffuse inter-reflections, caustics, dispersion within participating media, etc. It is also quite simple to implement and to extend in order to simulate new lighting phenomena.

Final gathering is the most time-consuming step of Photon Mapping. Based on the assumption of smooth geometry and thus, low-varying irradiance, caching methods [WRC88, Chr99] have been introduced to reduce the number of accesses to the Photon Map. As pointed out by Christensen and Batali [CB04] and Tabellion and Lamorlette [TL04], caching efficiency generally diminishes with the increasing geometric complexity of a 3D scene. Small surface details require a densely sampled cache in order to capture the subtle illumination variations induced by local variations of the normal (e.g., normal maps, micro-polygons, etc.).

In order to reduce this problem, this paper introduces a volumetric representation for indirect lighting based on irradiance vectors [Arv94]. From the irradiance vector, our representation inherits robustness against local variations of both photometric properties (diffuse component of reflectance) and geometric properties (surface normal vectors). Our representation therefore increases the efficiency of the

cache since fewer samples are required. From the volumetric structure and the associated interpolation scheme presented in this paper, we guarantee a smooth reconstruction everywhere in the 3D scene, making the cache representation directly accessible for the final rendering. Finally, our overall structure has low memory requirements and thus increases the scalability of the method. Furthermore, these qualities make this approach suitable for interactive hardware rendering.

This paper is organized as follows. After some related work, we introduce a global overview of our approach in Section 3. Then we detail how we construct the grid irradiance vectors with Photon Tracing in Section 4, and how we interpolate local irradiance for any 3D point in the scene in Section 5. We present and analyze some results, followed by a discussion of our approach and possible extensions.

## 2. Previous Work

Designing an appropriate representation of illumination is a difficult task since the range of possible effects can vary widely from low-frequency phenomena (e.g., diffuse reflections) to very high-frequency phenomena (e.g., sharp shadows, specular reflections, and caustics).

In a diffuse environment, radiosity algorithms (e.g., [SP94]) have proved their efficiency and their accuracy in





computing all light paths, while storing only one scalar value on each patch of the discretized scene. Unfortunately, when extended to higher-frequency phenomena, memory requirements explode: detailed shadows need fine tessellation or high-order bases [CAH00], directional distributions become necessary for glossy materials [SSG\*00], etc.

For high-frequency phenomena, stochastic approaches have been very efficient. Some of them, such as path tracing [Kaj86], bidirectional path tracing [LW93], and Metropolis light transport [VG97], are view-dependent and require very little intermediate storage. They can produce high-quality images without bias, but are generally time consuming. In order to reduce computation time, caching techniques have been introduced [WRC88].

*Photon Mapping* [Jen01] is currently the most popular solution to global illumination. Based on density estimation [WHSG97], particles (photons) are stored in different maps for different illumination phenomena (direct lighting, indirect lighting, caustics). During final gathering, a local search is performed in these maps for each intersection point in order to gather neighboring photons and estimate the local photon density. Despite its optimized structure (generally a kd-tree) and its efficient local search, this step is still time consuming, calling for caching [Chr99].

Caching precomputes a chosen radiometric quantity at different locations in the scene and interpolates between these values everywhere. Two main concerns with caching include: (1) photometric robustness (the kind of radiometric quantity); and (2) geometric robustness (how this quantity and the interpolation scheme scale when the geometric details increase).

On (1) photometric robustness, consider the following definition of the irradiance  $E(\mathbf{p}, \mathbf{n})$  at a point  $\mathbf{p}$  with normal  $\mathbf{n}$ :

$$E(\mathbf{p}, \mathbf{n}) = \int_{\Omega_{\mathbf{n}}} L(\mathbf{p} \leftarrow \boldsymbol{\omega}_i) \langle \mathbf{n}, \boldsymbol{\omega}_i \rangle d\boldsymbol{\omega}_i \quad (1)$$

where  $L(\mathbf{p} \leftarrow \boldsymbol{\omega}_i)$  represents the incident radiance at  $\mathbf{p}$  from

direction  $\boldsymbol{\omega}_i$ ,  $d\boldsymbol{\omega}_i$  the differential solid angle sustained by  $\boldsymbol{\omega}_i$ ,  $\Omega_{\mathbf{n}}$  the hemisphere centered at  $\mathbf{p}$  oriented toward  $\mathbf{n}$ , and  $\langle \cdot, \cdot \rangle$  denotes a dot product. This scalar quantity is theoretically material independent because it represents the integrated incident illumination. The reflected radiance  $L_r(\mathbf{p} \rightarrow \boldsymbol{\omega}_o)$ , necessary to compute an image, is material dependent and expressed as:

$$L_r(\mathbf{p} \rightarrow \boldsymbol{\omega}_o) = \int_{\Omega_{\mathbf{n}}} \rho(\mathbf{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L(\mathbf{p} \leftarrow \boldsymbol{\omega}_i) \langle \mathbf{n}, \boldsymbol{\omega}_i \rangle d\boldsymbol{\omega}_i \quad (2)$$

where  $\rho$  is the BRDF expression of the material. Under the assumption of diffuse BRDF  $\rho_D$ , Equation 2 reduces as below, using the precomputed irradiance:

$$L_r(\mathbf{p} \rightarrow \boldsymbol{\omega}_o) = \frac{\rho_D(\mathbf{p})}{\pi} E(\mathbf{p}, \mathbf{n}). \quad (3)$$

Therefore, irradiance caching schemes [WRC88, Chr99, CB04] allow the change of the diffuse albedo of a material.

To overcome the low BRDF frequency limitation, new schemes based on radiance caching have been introduced. Incident radiance is a scalar quantity integrated over a portion of the hemisphere. Spherical harmonics [AFO, KGPB05, KBPv06] or wavelets [CSSD94, SH94, SSG\*00] allow the efficient encoding of incident radiances than constant bases [GSHG92], but with both representations, the number of coefficients quickly explodes for high-frequency BRDFs, thus still limiting the method to low or glossy BRDFs.

Vector-based representations such as irradiance vectors [Arv94] are simpler and allow the representation of the same frequency range of BRDFs. They will form the basis of our representation. While light vectors [ZSP98] are also geometrically robust, they are not material independent, and therefore has less interesting to our application.

On (2) geometric robustness, as pointed by Equation 1, the irradiance depends on the surface normal and is therefore a geometric dependent quantity. Caching the radiance, theoretically removes this constraint. Unfortunately, the strategy [WRC88] used to place the precomputed samples depends on the underlying geometry. Therefore, radiance caching is not suitable when used with highly detailed surfaces, because the number of samples quickly becomes huge.

Finally, another issue with caching regards the interpolation scheme used during the rendering pass. Irradiance and radiance caching schemes need to store their samples in an efficient structure (kd-tree or octree) in order to quickly retrieve them when interpolation is needed. However, due to the combined facts that these samples are not placed on specific positions and that for efficiency reason, only an interpolation on local neighborhood is performed, these schemes cannot ensure a continuous reconstruction of the stored radiometric quantity. With a volumetric representation, such as the Irradiance Volume of Greger *et al.* [GSHG92], the continuous interpolation is easier to perform. Unfortunately,

their directional representation stores the incident radiance, requiring integration over the hemisphere for each normal in order to compute the corresponding irradiance.

We propose a novel method that stores volumetric irradiance vectors to improve on geometry robustness and results in similar same BRDF frequency as other comparable methods. To provide a continuous reconstruction of the irradiance vectors, we use a continuous interpolation scheme which does not depend on surface geometry. Furthermore, our representation is easy to adapt on GPU and has low memory consumption.

### 3. Overview

Our representation is based on a 3D regular grid aligned on the scene bounding box, subdivided at a user-specified resolution. A grid vertex (corner) stores one irradiance vector [Arv94] for each of the six main directions ( $\pm x | \pm y | \pm z$ ). Each irradiance vector captures the amount of incoming light, simulated by photon tracing [Jen01], traversing a face (rectangle) centered at the vertex. During a preprocessing step, the contribution of each photon is accumulated in the irradiance vectors associated with all oriented faces it traverses. Therefore photon hits are never explicitly stored.

During the final gathering at intersection point  $\mathbf{p}$  of surface normal  $\mathbf{n} = (n_x, n_y, n_z)$ , the irradiance vector at  $\mathbf{p}$  is interpolated from the irradiance vectors at the 8 voxel corners, and this, for the three directions corresponding to  $(n_x, 0, 0)$ ,  $(0, n_y, 0)$ , and  $(0, 0, n_z)$ . A final interpolation is performed on the unit sphere in order to compute the irradiance vector corresponding to direction  $\mathbf{n}$ .

## 4. Grid Construction

### 4.1. Irradiance Vector

The *irradiance vector*  $\mathbf{I}_n(\mathbf{p})$ , as introduced by Arvo [Arv94], is defined for a point  $\mathbf{p}$  with normal  $\mathbf{n}$  as

$$\mathbf{I}_n(\mathbf{p}) = \int_{\Omega_n} L(\mathbf{p} \leftarrow \boldsymbol{\omega}_i) \boldsymbol{\omega}_i d\boldsymbol{\omega}_i$$

and is directly related to the irradiance (cf. Equation 1) by

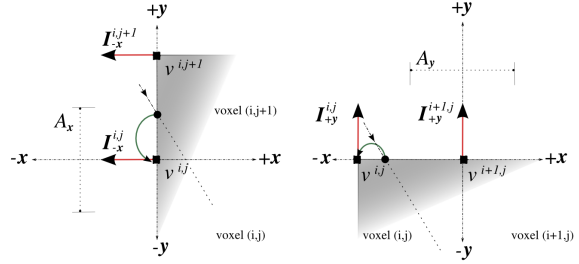
$$E(\mathbf{p}, \mathbf{n}) = \langle \mathbf{I}_n(\mathbf{p}), \mathbf{n} \rangle.$$

For a diffuse reflection (cf. Equation 3), the reflected radiance can be reformulated in terms of irradiance vector:

$$L_r(\mathbf{p} \rightarrow \boldsymbol{\omega}_o) = \frac{\rho_D(\mathbf{p})}{\pi} \langle \mathbf{I}_n(\mathbf{p}), \mathbf{n} \rangle. \quad (4)$$

Thus, for a local variation of the normal, the reflected radiance can be adjusted, making this representation more robust.

This value may also be easily reconstructed from the impacts of photons. Consider  $M$  photon hits on a surface element of area  $A$  and associated normal  $\mathbf{n}$ . If  $\phi_m$  is the energy



**Figure 1:** Irradiance Vector computation in 2D. When a photon hits a voxel's face its contribution is added to the irradiance vector associated with the nearest vertex. Photon contribution added to (Left)  $I_{-x}^{i,j}$  and (Right)  $I_{+y}^{i,j+1}$ .

of the  $m^{\text{th}}$  photon and  $\boldsymbol{\omega}_m$  its incident direction, the irradiance vector  $\mathbf{I}_n$  at this surface element is computed, using density estimation, as:

$$\mathbf{I}_n = \frac{1}{A} \sum_{m=1}^M \phi_m \boldsymbol{\omega}_m.$$

Since we use RGB colors, we have to store one irradiance vector for each color channel. This can be easily compressed if necessary, but in our software renderer, this never was an issue.

### 4.2. Grid of Irradiance Vectors

Our structure is based on an axis-aligned uniform rectangular 3D grid, divided into  $N_i \times N_j \times N_k$  voxels. At each vertex  $\mathbf{v}^{ijk}$  of the grid (where  $i \in [0, N_i]$ ,  $j \in [0, N_j]$ ,  $k \in [0, N_k]$ ), six irradiance vectors are stored, one for each main direction ( $\pm x | \pm y | \pm z$ ). In the remaining of this paper, we will note  $\mathbf{I}_{\boldsymbol{\delta}}^{ijk}$ , the irradiance vector stored at vertex  $\mathbf{v}^{ijk}$  in the direction  $\boldsymbol{\delta}$  where  $\boldsymbol{\delta} = \pm x | \pm y | \pm z$ .

### 4.3. Photon Propagation

To compute our irradiance vectors, we propagate photons through the grid. Every time a photon traverses a voxel face, its contribution is added to the irradiance vector  $\mathbf{I}_{\boldsymbol{\delta}}^{ijk}$  associated with the nearest vertex  $\mathbf{v}^{ijk}$  and the direction  $\boldsymbol{\delta}$  provided by the normal of the face (see Figure 1). When a photon hits some scene geometry, classical Russian roulette is applied and the photon is reflected according to its local BRDF. Once all photons have been treated, a normalization step is performed on the irradiance vectors:

$$\mathbf{I}_{\boldsymbol{\delta}}^{ijk} = \frac{1}{A_{\boldsymbol{\delta}}^{ijk}} \sum_{n=1}^{N_{\boldsymbol{\delta}}^{ijk}} \phi_n \boldsymbol{\omega}_n$$

where  $N_{\delta}^{ijk}$  is the number of photons that have contributed to the irradiance vector at vertex  $\mathbf{v}^{ijk}$  in direction  $\delta$ , and  $A_{\delta}^{ijk}$  is the area of the rectangular cell centered at  $\mathbf{v}^{ijk}$  in direction  $\delta$ . As we are using a uniform rectangular 3D grid, the area of such cell is simply the area of the voxel face oriented in the same direction, except for grid boundary vertices, where the area is divided by two, and for grid corner vertices, where it is divided by four. After this normalization step, our grid is ready for final gathering.

The photon propagation is accomplished in two steps. First, using a ray tracing acceleration structure [Gla89], we find the closest intersection with the scene. Then, we propagate the photon into the Irradiance Vector Grid without any intersection test.

Unlike the strategy of Havran *et al.* [HBHS05], our approach does not need to store all the rays generated from the photon propagation. Another benefit is that since we take into account the intersection of the photons with the grid for density estimation, our approach does not suffer of the classical boundary bias of Photon Mapping, where larger radii search spheres are used to collect photons, such as in room corners and along contours of flat surfaces.

## 5. Rendering with Indirect Illumination

### 5.1. Interpolation of Irradiance Vectors

In order to compute smooth values for irradiance vectors, we interpolate an irradiance vector for each point  $\mathbf{p}$  with normal  $\mathbf{n}$  that needs to be shaded (as in Equation 4). This interpolation is performed in two successive steps: a spatial interpolation according to  $\mathbf{p}$  and then a directional interpolation according to  $\mathbf{n}$ .

In the first step, the irradiance vector  $\mathbf{I}_{\delta}(\mathbf{p})$  at point  $\mathbf{p}$  in direction  $\delta$  is obtained by spatial interpolation of the irradiance vectors  $\mathbf{I}_{\delta}^{ijk}$  stored at the grid vertices surrounding point  $\mathbf{p}$ . The interpolation is only done for three out of the six possible directions of  $\delta$ . The choice between  $\pm\mathbf{x}$  (resp.  $\pm\mathbf{y}$  and  $\pm\mathbf{z}$ ) is done according to the sign of  $n_x$  (resp.  $n_y$  and  $n_z$ ).

In the second step, the final interpolated irradiance vector  $\mathbf{I}_{\mathbf{n}}(\mathbf{p})$  is simply obtained by remapping the three spatially interpolated irradiance vectors according to the normal direction  $\mathbf{n}$  at point  $\mathbf{p}$ :

$$\mathbf{I}_{\mathbf{n}}(\mathbf{p}) = \mathbf{I}_{\mathbf{x}}(\mathbf{p})n_x^2 + \mathbf{I}_{\mathbf{y}}(\mathbf{p})n_y^2 + \mathbf{I}_{\mathbf{z}}(\mathbf{p})n_z^2.$$

### 5.2. Direct Access to the Cache

Since our approach provides a smooth spatial and directional reconstruction of irradiance vectors, it can be used directly for diffuse or low-glossy indirect illumination: no need to propagate secondary rays for final gather.

Direct use of cached values can be done using previous techniques [WRC88, KBPv06], but they do not guarantee a

continuous reconstruction. Also, in order to provide a good quality solution, the sampling has to be adapted to the local variation of the geometry: highly-detailed geometry requires a large number of samples, reducing the overall efficiency.

Tri-linear or tri-cubic B-Spline interpolation provides satisfyingly smooth results for spatial interpolation of irradiance values in our software renderer. Each grid vertex therefore holds one irradiance vector  $\mathbf{I}_{\lambda}$  per color channel for each of the six  $\delta$  directions, i.e.,  $3 \times 3 \times 6 = 54$  floating point numbers, half of which must be accessed per interpolation. To reduce the costly texture fetches in current graphics hardware, we compress the 3 irradiance vectors as one color

$$r = \|\mathbf{I}_R\| \quad g = \|\mathbf{I}_G\| \quad b = \|\mathbf{I}_B\|$$

and one direction

$$\mathbf{I} = \frac{\mathbf{I}_R + \mathbf{I}_G + \mathbf{I}_B}{\|\mathbf{I}_R + \mathbf{I}_G + \mathbf{I}_B\|}.$$

These two vectors are encoded in two 16 bit 3D textures, and therefore the information for the six  $\delta$  directions require 12 3D textures. To efficiently perform tri-cubic interpolation in hardware, we adapted the technique of Sigg and Hadwiger [SH05].

### 5.3. Indirect Access to the Cache

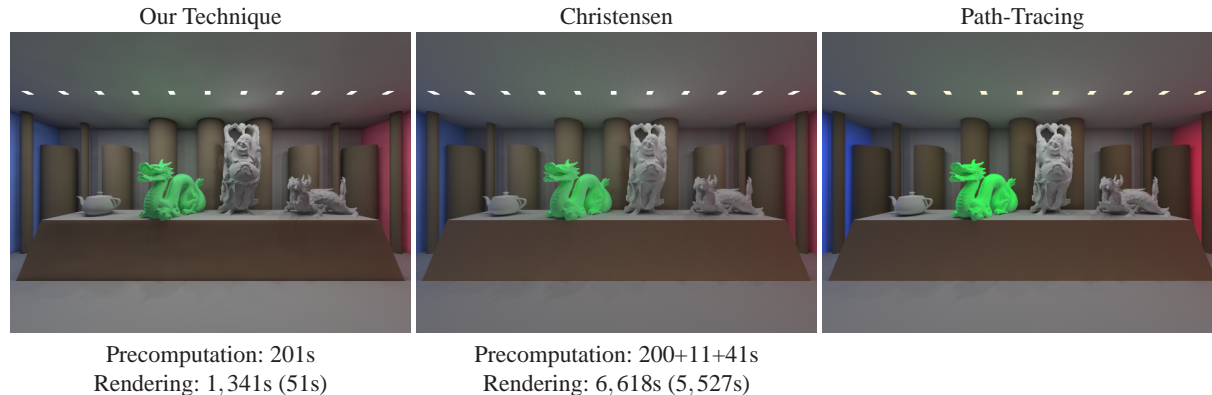
Classically in Photon Mapping, cached values are used indirectly: only secondary rays can access the cache. The main advantage is that high-frequency details such as indirect soft shadows are well preserved, and that reconstruction errors are masked since a diffuse or low-glossy reflection is similar to a low-pass filter [DHS\*05]. In this context, a simple tri-linear spatial interpolation provides a good compromise between speed and quality.

Note that in our approach, access time to our grid is constant in the number of the cached samples instead of logarithmic in the previous solutions [Chr99, CB04].

## 6. Results

To evaluate our approach, we present two test scenes with complex illumination conditions. The first configuration (Figure 2) presents a scene mainly directly illuminated from 11 light sources. The second one (Figure 3) is a classical two-room configuration where one room is indirectly illuminated by a light source placed in the other room. Both scenes have more than 8 million polygons due the highly detailed objects. They illustrate the geometric robustness of our approach.

We compare our technique against Photon Mapping combined with Christensen's [Chr99] precomputed samples. Precomputation time for our technique involves shooting photons and accumulating their contribution in the grid, and for Christensen's method involves shooting photons, balancing the kd-tree, and precomputing irradiance. For all images



**Figure 2:** Direct Illumination case. 16 rays per pixel were used. 5 M photons shot. (Left) Our technique with a  $40 \times 50 \times 40$  grid used directly with a tri-cubic interpolation scheme. (Center) Photon Mapping with Christensen’s Cache using 50 photons to precompute each irradiance sample. Precomputation is divided in: shooting photons + balancing the kd-tree + precomputing irradiance. (Right) Path-Tracing using 1600 rays per pixel. The general illumination patterns are similar in all three techniques. However our much faster indirect illumination computation (51s vs. 5,527s) strongly reduces the total rendering time (1,341s vs. 6,618s).

generated with Christensen’s method, we set the number of precomputed samples to be one fourth of the total number of photon hits, as originally suggested [Chr99].

All the results are computed on an AMD 64bit 3500+ processor with 2GB of memory. Images in Figure 2 and 3 have a  $640 \times 480$  resolution.

### 6.1. Direct Illuminated Scene

Figure 2 presents results of our technique for the direct lighting configuration. We provide a reference solution (right-most image) obtained with a classical Path-Tracing algorithm. For equivalent precomputation times, our technique allows a much faster computation (51s vs. 5,527s) of the indirect component of the illumination due to direct access to the cached values. While in our technique, the renderer spends most of its time computing ray-geometry intersections and direct illumination, Photon Mapping spends much more time shooting final gathering rays. Indirect access to the cached irradiance is necessary, since direct access produces very objectionable illumination patterns with zones of constant irradiance similar to Voronoi diagrams. This is a well-known artefact [Chr99] preventing direct access to cached irradiance.

### 6.2. Indirect Illuminated Scene

Our second test case is presented in Figure 3. For a direct access to our structure, and in order to capture finer shadows cast by the columns, a higher-resolution grid ( $30 \times 20 \times 92$ ) was built and consequently a larger number of photons needed to be shot. This resulted in a much larger precomputation time (2,270s). Still our method with direct access

is much faster (2,311s vs. 47,971s) than the one obtained with Christensen’s caching technique for a similar quality.

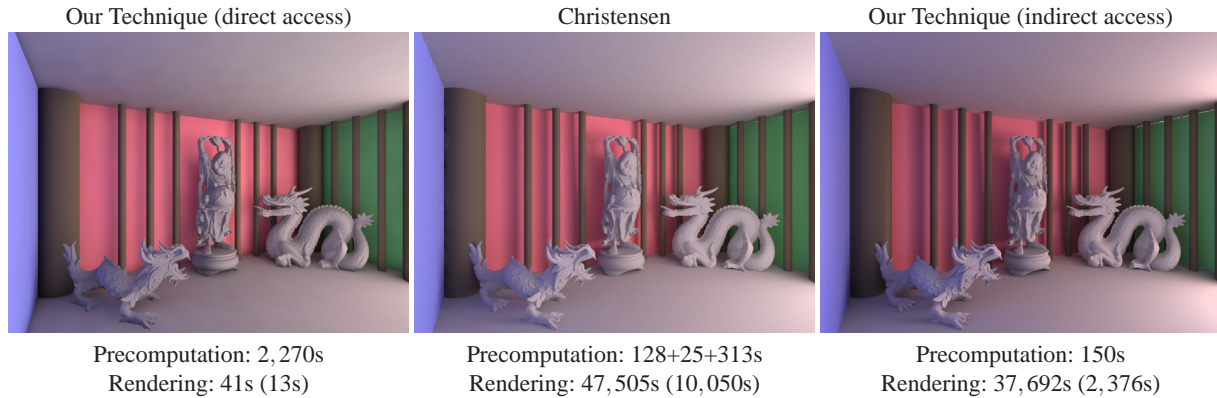
Another solution to capture the high frequency illumination phenomena consists in using our grid indirectly and shooting final gathering rays through it. In Figure 3 right, a lower resolution grid ( $12 \times 8 \times 20$ ) is traversed by the same number of photons than Christensen’s method (Figure 3 center). Our technique reduces both precomputation time (150s vs. 466s) and reconstruction time of the indirect illumination (2,376s vs. 10,050s), retaining the similar illumination features. The direct access to our simple grid instead of a kd-tree explains these gains.

### 6.3. Geometric Robustness

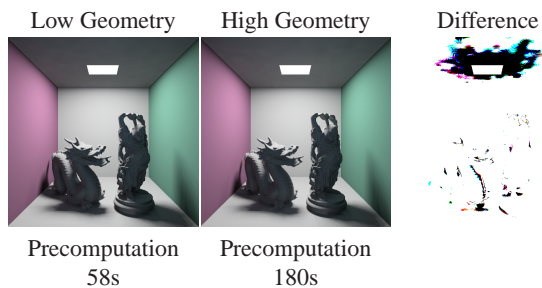
Due to its robustness against geometric details, it is possible with our approach to precompute the volumetric irradiance vectors on a low resolution of the objects, similarly to the approach developed by Tabellion and Lamorlette [TL04]. In Figure 4, the same grid resolution and number of photons (8 M) are used during the precomputation. The resulting illuminations recomputed over the two images are quite similar, for a lower precomputation time. Changing the geometry refines the curvature of the objects and therefore modifies the photon reflection, explaining the little differences located on the two objects as on the ceiling.

### 6.4. Indirect Illumination on Hardware

Figure 5 shows the quality and framerates of our reconstruction when using 3D textures on the GPU (NVIDIA 7900 GTX at a  $600 \times 600$  resolution). We precomputed a

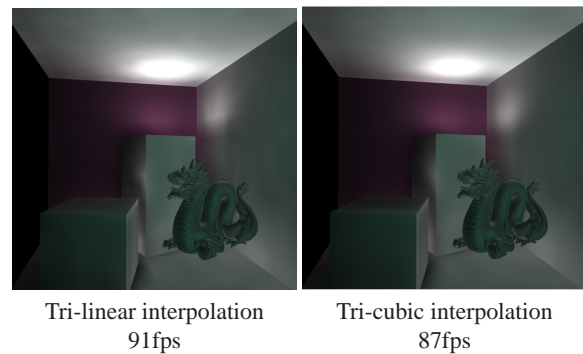


**Figure 3:** Indirect illumination case. (Left) Our technique using a  $30 \times 20 \times 92$  grid with 80 M photons. (Center) Christensen’s method reference image with 3200 rays to sample the hemisphere. (Right) Our technique using indirectly a  $12 \times 8 \times 20$  grid with the same number of photons shot and the same number of rays to sample the hemisphere, than Christensen’s method. Direct access to our grid provides very fast rendering but the grid resolution might need to be increased, forcing more expensive precomputations. Indirect access to our grid results in a more expensive rendering but in a lower precomputation costs. Both solutions are available, with their advantages and disadvantages.



**Figure 4:** Geometric robustness illustration when using direct access. The precomputation is performed on a  $20 \times 22 \times 20$  grid using (Left) low detailed geometry (23,650 triangles) and (Center) high detailed geometry (1,952,000 triangles). (Right) The difference color-coded image between the two reconstructions (white: no difference, black: difference of RGB value  $(\Delta r + \Delta g + \Delta b) / (3 \times 255)$ , maximum pixel difference is  $11 / (3 \times 255)$ ).

$16 \times 16 \times 16$  grid without the dragon inside a color modified Cornell box. The two images show, using (left) a tri-linear and (right) a tri-cubic interpolation, the indirect illumination (i.e., irradiance multiplied by the diffuse albedo) on the dragon and all scene objects. This approach is mostly geometry independent since the dragon (around 400,000 triangles) is moved interactively while being coherently indirectly illuminated. The cost of using our structure is pixel dependent, since the lighting evaluation is performed on the fragment/pixel processor.



**Figure 5:** Reconstruction on the GPU. The camera and the dragon can be moved at the indicated frame rates.

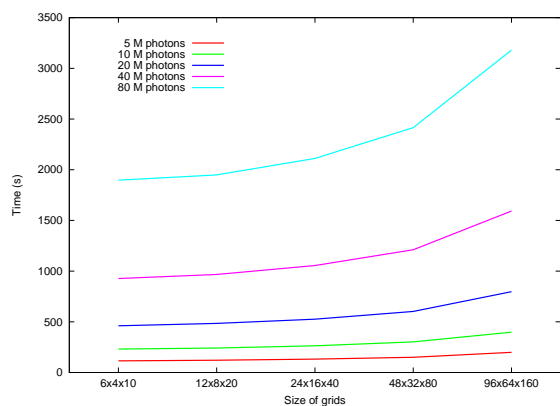
## 6.5. Discussion/Influence of the Parameters

Figure 7 shows the construction time of our grid for different numbers of grid resolutions and photons shot. The construction time grows mainly linearly with the number of photons and a small overhead exists when increasing the size of the grid.

As shown in Figure 6, even with a still grid size and fewer photons than in Figure 3, our structure is still able to capture the indirect effect of the light near geometrically complex objects. Noise due to undersampling of the illumination during the photon tracing pass appears mostly on larger planes and on higher frequency details such as shadows. The quality of the reconstruction scales directly with the number of photons shot, as the illumination details scale directly with the size of the grid.



**Figure 6:** A direct reconstruction using the  $24 \times 16 \times 40$  grid with 5 M photons.



**Figure 7:** Construction times as a function of grid resolutions and number of photons shot.

In certain configurations, our approach can suffer from limitations, mostly related to the regularity of our grid structure. First, even if un-biased, our estimation can lead to energy leaks or under-estimation of irradiance values, depending on the relative geometric configuration with the grid. Our approach cannot reach the reconstruction quality of Ray maps [HBHS05], yet less storage is required. These artefacts are reduced by increasing the resolution of the grid. A hierarchical and adaptive grid would reduce the required memory.

The regular grid can also limit illumination to low-frequency effects: as in Photon Mapping, the quality of the details depends on the chosen filter, which is constant in our case. Once again, a hierarchical and adaptive grid would help in preserving smooth features such as the softness of shadows and sharp features such as caustics.

## 7. Conclusion

In this paper, we have presented a new representation for indirect illumination, based on a 3D grid of irradiance vectors. This representation allows a smooth-everywhere reconstruction of the irradiance. Thanks to the irradiance vectors, the resulting solution is more robust to local variation of geometry and photometric properties, as shown through the presented results.

We have implemented this approach as a caching scheme for Photon Mapping. The results show that diffuse inter-reflections are well captured. Compared to existing solutions, our approach requires fewer cached samples for higher-quality cached indirect lighting. It does not also require any photon storage. Furthermore, our irradiance cache can be directly accessed during the final gathering pass.

Our hardware implementation demonstrates how it can be suitable for quality real-time display of 3D scenes under pre-computed global illumination.

## Future Work

There are two main directions for a quality improvement of our approach. Firstly, the estimation of irradiance vectors is based on a regular grid. Identifying proper resolutions for the grid and the number of photons to shoot requires a good understand of the illumination. A more automatic estimate would facilitate the use of the technique. For large scenes, we would like to use a multiresolution and adaptive structure in order to reduce the cost of our grid's construction. This will also allow the more precise capture of the indirect illumination near surfaces and maybe find better automatic estimates.

Secondly, since the scheme is based on linear interpolation, some artefacts can appear for more complex variations of illumination. The introduction of gradients would improved the smoothness of the reconstruction where incoming illumination varies quickly. Improve visibility estimation will also provide higher-quality reconstruction.

## References

- [AFO] ARIKAN O., FORSYTH D., O'BRIEN J.: Fast and Detailed Approximate Global Illumination by Irradiance Decomposition. In *Proc. SIGGRAPH 2005*, ACM, pp. 1108–1114.
- [Arv94] ARVO J.: The irradiance Jacobian for partially occluded polyhedral sources. In *Proc. SIGGRAPH'94* (1994), ACM, pp. 343–350.
- [CAH00] CUNY F., ALONSO L., HOLZSCHUCH N.: A novel approach makes higher order wavelets really efficient for radiosity. *Computer Graphics Forum* 19, 3 (2000), 99–108.
- [CB04] CHRISTENSEN P., BATALI D.: An Irradiance Atlas for Global Illumination in Complex Production Scenes. In *Proc. Eurographics Symposium on Rendering* (2004), pp. 133–141.

- [Chr99] CHRISTENSEN P.: Faster global photon map global illumination. *Journal of Graphics Tools* 4, 3 (1999), 1–10.
- [CSSD94] CHRISTENSEN P., STOLLNITZ E., SALESIN D., DE ROSE T.: Wavelet radiance. In *Proc. Eurographics Workshop on Rendering* (1994), pp. 287–302.
- [DHS\*05] DURAND F., HOLZSCHUCH N., SOLER C., CHAN E., SILLION F.: A frequency analysis of light transport. *ACM Trans. on Graph.* 24, 3 (2005).
- [Gla89] GLASSNER A. S.: *An Introduction to Ray Tracing*. Morgan Kaufmann, 1989.
- [GSHG92] GREGER G., SHIRLEY P., HUBBARD P., GREENBERG D.: Irradiance volume. *IEEE Computer Graphics and Applications* 18, 2 (1992), 32–43.
- [HBHS05] HAVRAN V., BITTNER J., HERZOG R., SEIDEL H.-P.: Ray Maps for Global Illumination. In *Proc. Eurographics Symposium on Rendering 2005* (2005), pp. 43–54.
- [Jen01] JENSEN H. W.: *Realistic Image Synthesis using Photon Mapping*. A.K. Peters, 2001.
- [Kaj86] KAJIYA J.: The rendering equation. In *Proc. SIGGRAPH'86* (1986), pp. 143–150.
- [KBPv06] KRÍVÁNEK J., BOUATOUCH K., PATTANAIK S., ŽÁRA J.: Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. In *Proc. Eurographics Symposium on Rendering 2006* (2006), pp. 127–138.
- [KGPB05] KRÍVÁNEK J., GAUTRON P., PATTANAIK S., BOUATOUCH K.: Radiance caching for efficient global illumination computation. *IEEE Trans. Visualization and Computer Graphics* 11, 5 (2005), 550–561.
- [LW93] LAFORTUNE E., WILLEMS Y.: Bi-directional path tracing. In *Proc. Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)* (1993), pp. 145–153.
- [SH94] SCHRÖDER P., HANRAHAN P.: Wavelet methods for radiance computations. In *Proc. Eurographics Workshop on Rendering* (1994), pp. 310–326.
- [SH05] SIGG C., HADWIGER M.: *Fast Third-Order Texture Filtering in GPU Gems2*. Addison-Wesley, 2005.
- [SP94] SILLION F. X., PUECH C.: *Radiosity and Global Illumination*. Morgan Kaufmann, 1994.
- [SSG\*00] STAMMINGER M., SCHEEL A., GRANIER X., PEREZ-CAZORLA F., DRETTAKIS G., SILLION F.: Efficient glossy global illumination with interactive viewing. *Computer Graphics Forum* 19, 1 (2000), 13–25.
- [TL04] TABELLION E., LAMORLETTE A.: An approximate global illumination system for computer generated films. *ACM Trans. on Graphics* 23, 3 (2004), 469–476.
- [VG97] VEACH E., GUIBAS L.: Metropolis light transport. In *Proc. SIGGRAPH'97* (1997), pp. 65–76.
- [WHS97] WALTER B., HUBBARD P., SHIRLEY P., GREENBERG D.: Global illumination using local linear density estimation. *ACM Trans. on Graphics* 16, 3 (1997), 217–259.
- [WRC88] WARD G., RUBINSTEIN F., CLEAR R.: A ray tracing solution for diffuse interreflection. In *Proc. SIGGRAPH'88* (1988), pp. 85–92.
- [ZSP98] ZANINETTI J., SERPAGGI X., PÉROCHE B.: A vector approach for global illumination in ray tracing. In *Proc. Eurographics 1998* (1998), pp. 149–158.