N° d'ordre : 3989

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par **Jérôme Baril**

Pour obtenir le grade de

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Modèles de représentation multi-résolution pour le rendu photo-réaliste de matériaux complexes

---

Soutenue le : 11 Janvier 2010

Après avis des rapporteurs :

Laurent Grisoni ..	Professeur
Mathias Paulin ...	Professeur

Devant la commission d'examen composée de :

Pascal Guitton ...	Professeur	Président
Laurent Grisoni ..	Professeur	Rapporteur
Mathias Paulin ...	Professeur	Rapporteur
Christian Bouville	Professeur	Examinateur
Christophe Schlick	Professeur	Directeur de Thèse
Patrick Gioia	Ingénieur de Recherche	Co-Directeur de Thèse

– 2010 –

Remerciements

Je tiens tout d'abord à remercier mes encadrants de thèse, Christophe Schlick et Patrick Gioia, sans qui les travaux réalisés durant cette thèse n'auraient jamais aboutis. Je remercie Christophe pour sa gentillesse, ses qualités d'écoute et de compréhension, sa rigueur scientifique, ses qualités pédagogiques et pour toutes les longues discussions contructives que nous avons toujours eu tout a long de ma thèse. Je remercie Patrick pour m'avoir accepté au sein de son équipe de recherche, ses explications mathématiques, son exigeance scientifique et pour avoir supporté mon carractère sans jamais vaciller.

Je remercie les deux rapporteurs de mon manuscrit de thèse, Mathias Paulin et Laurent Grisoni, pour leur intérêt porté aux travaux présentés dans ce document, leur capacité d'analyse et leur expertise dont ils ont fait preuve dans les rapports de thèse. Je remercie également Christian Bouville qui a bien voulu être l'examinateur de mon manuscrit de thèse. J'ai beaucoup apprécié toutes les discussions que nous avons eu durant ma thèse ainsi que tous les conseils et suggestions qu'il m'a fourni dans ses commentaires sur mes travaux de thèse.

Je remercie le LaBRI et l'INRIA ainsi que Orange Labs pour avoir financé mon doctorat sans restriction. Tout ça n'aurait pas été possible sans Pascal Guitton et Danielle Pelé qui ont bien voulu m'accepter dans leur laboratoire respectif. Un merci tout particulier à Pascal Guitton pour son soutien sans faille durant toutes les années que j'ai passé au sein de l'équipe IPARLA ainsi que pour l'honneur qu'il m'a fait en acceptant de présider mon jury de thèse.

Beaucoup de personnes m'ont encouragé, supporté ou aidé pendant les longues heures de travail au laboratoire. Du coté rennais, je remercie Nicolas Ramin pour sa gentillesse, nos discussions et pour sa présence lors de nos traditionnelles pauses syndicales. Ensuite, merci à Julien, Jérôme, Raphaele, Samuel et bien d'autres encore pour avoir apporté du soleil dans le ciel breton. Du coté bordelais, je remercie Tamy Boubekeur pour m'avoir soutenu,

encouragé et guidé pendant tout le début de ma thèse. Merci à Pac, Romain, Tiz, Elisabeth, David, Pascal, Xavier, Martin et plus généralement à toute l'équipe Iparla pour leur bonne humeur et leur amitié. Un merci particulier à toutes les personnes qui ont bien voulu relire mon manuscrit de thèse !

D'un point de vue personnel, je tiens à remercier tous mes amis, en vrac, Cyril, Lolo, Candice, les grumeaux, Pac, Romain, Nicolas, Julien, Marie, Nathalie, Jérôme, Maricha, Alexis, Nadiège, Mathieu, Emilie, David, AnneSo, Ronan, Blandine, et tous ceux que j'ai oublié pour tous les bons moments passés ensemble durant cette thèse. Enfin, le dernier mais pas le moindre, je remercie énormément toute ma famille sans qui je ne serais sans doute jamais arrivé là et plus particulièrement mes parents, ma soeur, Pascal et Gabin. Un gigantesque merci à ma p'tite soeur qui a relu plusieurs fois ce manuscrit malgré sa réticence pour le domaine de l'informatique !

Merci (et désolé) à tous ceux que je n'ai pas mentionné ici mais qui ont fait partie de ma vie durant ces années de thèse !

Résumé

L'émergence des périphériques de capture numériques ont permis le développement de l'acquisition 3D pour numériser les propriétés d'un objet réel : sa forme et son apparence. Ce processus fournit une représentation dense et précise d'objets réels et permet de s'abstraire d'un processus de simulation physique coûteux pour modéliser un objet. Ainsi, les problématiques ont évolué et portent non plus uniquement sur la modélisation des caractéristiques d'un objet réel mais sur les traitements de données issues de l'acquisition pour intégrer une copie de la réalité dans un processus de synthèse d'images. Dans ces travaux de thèse, nous proposons de nouvelles représentations pour des fonctions d'apparence issues de l'acquisition dont le but est de définir un ensemble de modèles multi-échelles, de faible complexité en taille, capable d'être visualisé en temps réel sur le matériel graphique actuel.

Mots-clés : informatique graphique, synthèse d'images, rendu, matériaux, rendu temps-réel, apparence, multi-résolution, ondelettes, GPU, BRDF, SVBRDF, BTF, PTM, approximation.

Abstract

The emergence of digital capture devices have enabled the development of 3D acquisition to scan the properties of a real object : its shape and its appearance. This process provides a dense and accurate representation of real objects and allows to avoid the costly process of physical simulation to model an object. Thus, the issues have evolved and are no longer focus on modeling the characteristics of a real object only but on the treatment of data from acquisition to integrate a copy of reality in a process of image synthesis. In this thesis, we propose new representations for appearance functions from the acquisition with the aim of defining a set of multiscale models of low complexity in size working in real time on the today's graphics hardware.

Keywords : computer graphic, image synthesis, rendering, material, real time rendering, appearance, multi-resolution, wavelets, GPU, BRDF, SVBRDF, BTF, PTM, approximation.

Table des matières

Introduction et Motivations **1**

I Domaines de recherche **5**

1 Apparence d'une surface **7**
1.1 Apparence et multi-échelle 8
1.2 Modèles de réflectance 10
1.2.1 Matériaux uniformes 11
1.2.2 Matériaux non-uniformes 13

2 *Bidirectional Texture Functions* **17**
2.1 Acquisition 19
2.2 Compression 24
2.2.1 Compression basée approximation 26
2.2.2 Compression basée factorisation 31
2.2.3 Compression par substitution 37
2.2.4 Bilan 38
2.3 Edition et Synthèse 40

3 Les ondelettes **43**
3.1 Ondelettes et analyse multi-résolution 43
3.2 Cas pratiques 45
3.2.1 Ondelettes de première génération 45
3.2.2 Ondelettes géométriques 47
3.3 Compression par ondelettes 49
3.3.1 Intuition 49
3.3.2 Codage *Zero-Tree* 51
3.3.3 Codages typés matériel graphique 53

II Modèles et représentations proposés **57**

4 Ondelettes et *BTF* : Premières expérimentations **59**
4.1 Mise en place d'une *BTF* multi-échelle 59
4.2 Codage *Zero-Tree* 61
4.2.1 Décomposition en ondelettes 62
4.2.2 Quantification 63
4.2.3 Codage 64
4.3 Résultats et discussion 65

5 Arbre d'ondelettes polynômial **71**
5.1 Présentation 71
5.2 Approximation polynomiale 72
5.2.1 Choix de la base d'approximation 72
5.2.2 Paramétrisation de l'espace des directions 74
5.2.3 Résultats sur l'erreur d'approximation 76
5.3 Simplification adaptative 78
5.3.1 Paquets d'ondelettes polynomiaux 78
5.3.2 Prédiction d'erreur 80
5.3.3 Résultats de la simplification adaptative 82
5.4 Rendu 84
5.5 Discussion et approches futures 86

6 Hachage et arbres d'ondelettes **89**
6.1 Transformation multi-échelle 90
6.2 Quantification adaptative 93
6.2.1 Critère qualitatif 95
6.2.2 Quantification 96
6.3 Hachage des coefficients 98
6.3.1 Présentation 98
6.3.2 Carte du domaine 100
6.3.3 Hachage des coefficients 102
6.4 Résultats 105
6.5 Bilan et approches futures 109

Conclusion **113**
Résumé des contributions 113
Perspectives 114

x	$\in \mathbb{R}^2$	Coordonnées (u, v) d'un pixel dans une texture
(θ, ϕ)	$\in \mathbb{S}^2$	Direction en coordonnées polaire. θ représente l'élévation et ϕ l'azimut
$\vec{\omega}_i$	$\in \mathbb{S}^2$	Direction incidente de la lumière en coordonnées polaire (θ_i, ϕ_i)
$\vec{\omega}_r$	$\in \mathbb{S}^2$	Vecteur de reflexion en coordonnées polaire (θ_r, ϕ_r)
$\vec{\omega}_o$	$\in \mathbb{S}^2$	Direction sortante du point de vue d'observation en coordonnées polaire (θ_o, ϕ_o)
η_x	$\in \mathbb{N}^2$	Nombre de pixels dans une texture $(\eta_u \times \eta_v)$
η_i	$\in \mathbb{N}$	Nombre d'échantillons des directions de la lumière
η_o	$\in \mathbb{N}$	Nombre d'échantillons des directions du point de vue
$\mathbb{T}$	$\subset \mathbb{R}^2$	Ensemble des pixels d'une texture
$\mathbb{W}_i$	$\subset \mathbb{S}^2$	Ensemble des échantillons des directions de la lumière
$\mathbb{W}_o$	$\subset \mathbb{S}^2$	Ensemble des échantillons des directions du point de vue
		Représentation de la projection paraboloïd des vecteurs $\vec{\omega}_o$ et $\vec{\omega}_i$. Ici, $(\theta_o, \phi_o) = (0; 0)$ et $(\theta_i, \phi_i) = (30; 90)$

BRDF ..... *Bidirectional Reflectance Distribution Function.*

BSSRDF ..... *Bidirectional Subsurface Scattering Reflectance Distribution Function.*

SVBRDF ..... *Spatially-Varying Bidirectional Reflectance Distribution Function.*

BTF ..... *Bidirectional Texture Functions.*

1D, 2D, 3D, 4D... une, deux, trois, quatre dimensions.

$< \vec{a}, \vec{b} >$ ..... Produit scalaire entre le vecteur $\vec{a}$ et le vecteur $\vec{b}$.

Introduction et Motivations

L'informatique graphique est un domaine en plein essor de par les besoins grandissants de visualisations d'images générées à partir de méthodes numériques. De nombreux secteurs, industriels ou scientifiques, participent à l'expansion de cette activité. L'industrie du film et du jeu vidéo ou encore l'imagerie médicale en sont les principaux moteurs à travers une recherche perpétuelle de méthodes de visualisation innovantes. Suivant les besoins, le rendu d'une scène virtuelle peut être calculé suivant différentes approches. D'un coté, des effets de stylisation sont utilisés pour représenter un scène de manière schématique ou artistique. Ces techniques sont le plus souvent exploitées pour clarifier les différentes composantes d'une scène, reproduire un effet artistique ou encore mettre en avant les spécificités d'un objet. L'ensemble de ces méthodes de rendu est regroupé sous la thématique du rendu expressif. D'un autre coté, l'immersion d'un ensemble de participants dans un univers physiquement plausible est un critère de motivation pour les méthodes dites de rendu réaliste. Les images générées par des méthodes de calculs complexes ont atteint un degré de réalisme capable de tromper la perception (cf. Figure 1). Le temps de calcul pour produire une image de synthèse pour laquelle le degré de réalisme est à son paroxysme est de l'ordre de quelques heures. Cependant, en effectuant des simplifications physiques, ce traitement peut se faire de nos jours en temps réel grâce à l'évolution des performances du matériel graphique. La qualité du rendu en temps réel tend à se rapprocher des méthodes de rendu hors-ligne les plus abouties mais, une des difficultés principales pour ce type de rendu réside dans le respect des spécificités d'un objet et de ses interactions avec son environnement naturel dans une scène de synthèse.

Les avancées technologiques sur les périphériques de capture numériques ont permis le développement du domaine de l'acquisition 3D. Le but de cette approche est de numériser le plus fidèlement possible les caractéristiques d'un objet réel pour l'intégrer dans une scène virtuelle. Les scanners 3D, les spectromètres ou encore les appareils photos numériques sont les périphériques de

capture privilégiés pour acquérir toutes les propriétés d'un objet : sa forme et son apparence. Cette numérisation de la réalité donne une représentation dense et précise d'objets réels et permet de s'abstraire d'un processus de simulation physique coûteux pour reproduire la forme et l'apparence d'un objet. Ainsi, des thèmes de recherche ont émergé, non plus sur la modélisation des caractéristiques d'un objet réel, mais sur les traitements de données issues de l'acquisition dans le but d'intégrer une copie de la réalité au sein d'un environnement virtuel.



(a) (b)

FIGURE 1: Comparaison entre une image réelle et une image de synthèse : (a) une image réelle, (b) une image de synthèse générée à l'aide du moteur de rendu ©CryEngine2.

Les travaux présentés dans ce manuscrit s'inscrivent dans ce contexte et, plus particulièrement, sur le traitement de fonctions d'apparence issues de l'acquisition. En effet, la multitude d'informations générées lors du processus de numérisation d'un matériau nécessite un ensemble de traitements pour être intégrée dans une scène virtuelle. La limitation principale réside dans la taille des données dans la mesure où la quantité nécessaire au stockage d'un matériau brute post-acquisition est de l'ordre de quelques giga-octets. Une deuxième problématique repose sur la construction d'un flux suffisamment compacte mais aussi flexible pour permettre un rendu en temps réel de ces fonctions d'apparence sur le matériel graphique actuel.

Les axes de recherche étudiés dans ce mémoire de thèse sont donc regroupés suivant deux thèmes principaux. Premièrement, nous avons développé une étude sur des matériaux issus de l'acquisition à travers les « Bidirectional Texture Functions » (*BTFs*). Le premier thème de recherche, à caractère applicatif, se positionne donc sur la mise en place de structures de données sur ce type de matériaux permettant un rendu efficace sur matériel graphique. L'outil mathématique privilégié dans ce contexte est la transformation en ondelettes. De part cette utilisation, un deuxième thème de recherche, plus général, a conduit à nous interroger sur des méthodes de stockage efficaces pour des signaux en ondelettes capables d'être utilisées sur le matériel graphique actuel.

Ainsi, les contributions apportées dans ces travaux de thèse sont, en

premier lieu, une analyse des différentes méthodes mathématiques pouvant être utilisées sur les *BTFs*. Cette étude nous a permis de mettre en place des algorithmes de compression efficaces sur ce type de données à travers des outils de manipulation multi-échelles tout en respectant la contrainte de portabilité sur le matériel graphique. Enfin, dans une approche préliminaire, nous tâchons de mettre en place les étapes nécessaires à la construction d'un nouveau standard de compression capable d'être stocké de manière compacte en mémoire vidéo et ainsi d'être reconstruit en temps réel sur les processeurs graphiques programmables.

Ce mémoire est organisé en deux grandes parties. La première partie (Chapitres 1 à 3) traite des domaines de recherche impliqués dans la mise en place de nos travaux. La deuxième partie (Chapitres 4 à 6) expose les différentes méthodes développées pour la compression de *BTFs* et la mise en place d'un nouveau standard de compression.

Après avoir rappelé au Chapitre 1 les principes généraux sur l'apparence d'un objet, nous étudions dans le Chapitre 2, les spécificités des *BTFs* ainsi que les différentes approches de traitement existantes sur ce type de matériau. Par la suite, dans le Chapitre 3, nous définissons les mécanismes mis en jeux dans une analyse en ondelettes à travers une explication théorique suivie de cas pratiques en abordant les méthodes de l'état de l'art sur la compression de signaux en ondelettes.

Le Chapitre 4 traite des premières expérimentations que nous avons effectuées pour valider l'approche basée ondelettes sur des matériaux tels que les *BTFs*. Dans le Chapitre 5, nous introduisons une méthode de compression de *BTFs* permettant de fournir une représentation progressive de tels matériaux qui autorise un rendu en temps réel sur le matériel graphique. Enfin, dans le Chapitre 6, nous développons un ensemble d'étapes permettant la mise en place d'un standard de compression à travers le domaine applicatif des *BTFs* mais dont on fournit une généralisation sur le traitement de signaux en ondelettes.

Pour chaque partie, nous procédons une analyse des méthodes exposées en clarifiant les limitations des différentes approches et en proposant des travaux futurs permettant d'améliorer ces méthodes.

Première partie

Domaines de recherche

Chapitre 1

Apparence d'une surface

« *On ne fait pas la lumière, on la reproduit.* »
Paul Cézanne

Pour définir l'apparence d'une surface, la notion la plus élémentaire est la couleur. La couleur d'une surface est le résultat d'un échange énergétique entre une source lumineuse incidente et la surface dont les propriétés de réflexion varient suivant la matériau considéré. Les propriétés du matériau déterminent la quantité d'énergie réfléchie ou absorbée pour chaque longueur d'onde constituant le spectre visible de la lumière. C'est pourquoi, en informatique graphique, le terme de *réflectance*, au sens large, est privilégié pour décrire l'apparence d'une surface plutôt que la couleur. La réflectance d'une surface est une notion complexe à modéliser à cause des nombreux phénomènes physiques qu'elle peut représenter. La Figure 1.1 décrit le comportement général d'un rayon lumineux en contact avec une surface.

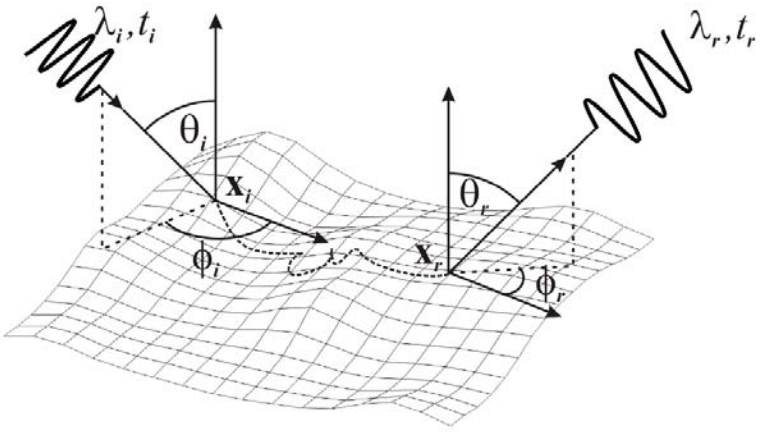


Figure 1.1: Trajet d'un rayon lumineux incident sur une surface.

Un rayon lumineux de longueur d'onde λ_i frappe la surface en un point

x_i au temps t_i. Dans le repère local au point x_i, la direction du rayon lumineux incident est (θ_i, ϕ_i) (en coordonnées polaires). Le rayon lumineux se propage le long de la surface pour être reflété au point x_r au temps t_r. De part l'influence du matériau de la surface, le rayon lumineux reflété peut avoir une longueur d'onde différente ($\lambda_r \neq \lambda_i$) engendré par la fluorescence, l'irridescence, ou la phosphorescence. Le vecteur de réflexion, exprimé dans le repère local au point x_r, est noté (θ_r, ϕ_r). La fonction qui modélise l'interaction entre un rayon lumineux et une surface est donc une fonction à 12 dimensions.

En pratique, les méthodes numériques pour résoudre de telles équations ne sont pas envisageables car la complexité dimensionnelle est trop élevée. Des simplifications sont donc à effectuer pour optimiser les calculs tout en restant visuellement réaliste. De plus, d'autres éléments « perturbateurs » doivent être pris en compte pour caractériser l'apparence d'une surface. Ainsi, dans la suite de ce chapitre, nous allons, dans un premier temps, définir l'apparence d'une surface suivant plusieurs échelles et énumérer ainsi les paramètres les plus influents pour la modélisation d'un matériau. Par la suite, nous présenterons les modèles les plus représentatifs de la littérature pour modéliser l'apparence.

1.1 Apparence et multi-échelle

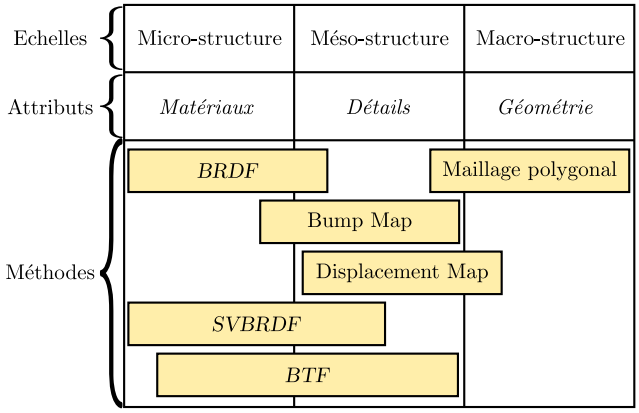


FIGURE 1.2: Propriétés d'une surface à différentes échelles et représentations usuelles correspondantes.

Les propriétés d'apparence d'une surface sont souvent formalisées suivant l'échelle à laquelle on se place [McA02] (voir Figure 1.2). Le niveau le plus grossier, macroscopique, fait référence à la forme de l'objet, sa géométrie. Une représentation conventionnelle en informatique graphique consiste à traiter la géométrie comme un ensemble de polygones reliés les uns aux autres pour former un maillage. Ce paradigme découle des propriétés du matériel graphique

dont la primitive d'entrée privilégiée est le triangle.

La micro-structure caractérise la relation qui existe entre un matériau et la lumière : une surface en bois ne va pas réfléchir la lumière de la même manière qu'une surface métallique. Elle est définie généralement par la *Bidirectional Reflectance Distribution Function* (*BRDF*) [Nic70], fonction de plusieurs variables qui détermine la quantité de lumière réfléchie à la surface pour une source de lumière incidente et un point de vue (i.e. la réflectance). Une grande variété de matériaux peut être représentée par une telle fonction. Cependant, les matériaux translucides, par exemple, ont des propriétés de réflexion plus complexes et nécessitent des paramètres supplémentaires pour être modélisés (voir Section 1.2.1).



(a) (b)

FIGURE 1.3: La Lune représentée par (a) sa forme et une couleur en chaque point (b) agrémentée par sa méso-structure pour faire ressortir les détails à la surface (*Bump Mapping* [Bli78]).

La méso-structure représente un niveau de détails intermédiaire. Elle encapsule les variations géométriques infimes propres à un matériau. En informatique graphique, de nombreuses méthodes sont disponibles dans la littérature pour caractériser ces structures. Pour de faibles variations géométriques, la méthode la plus utilisée est le « *Bump Mapping* » [Bli78] : une perturbation des normales à la surface pour donner l'illusion de micro-relief. Un exemple est donnée sur la Figure 1.3 où la méso-structure permet de faire ressortir les cratères à la surface de la lune. Pour des variations géométriques plus importantes, le « *Displacement Mapping* » [CT82] est privilégié où le principe consiste en une variation géométrique des points du maillage pour modéliser un faible relief. Cette échelle de détails est donc un niveau intermédiaire entre la géométrie et la manière dont la lumière va interagir avec la surface.

De par la complexité des surfaces réelles, il est difficile de déterminer une représentation fidèle à la réalité en traitant le problème individuellement par échelle : la méso- et la micro-structure sont indissociables. En



(a) (b)

FIGURE 1.4: Une scène réaliste rendu à l'aide (a) des normales uniquement et (b) des normales et d'une surface d'élévation par la technique de *Parallax Mapping* définie dans [Tat06]

effet, les variations géométriques locales de la surface conduisent à des variations locales de la lumière (inter-réflexions, ombrage) mais aussi à des phénomènes de masquage (occultations). Un certain nombre de méthodes [POC05, PO06, Tat06, WWT+03] traite de ces problématiques en approximant les effets de parallaxes, induits par un changement de point de vue, par l'ajout d'une carte d'élévation correspondant à la méso-structure de la surface d'un objet. Un lancé de rayon est effectué à travers la texture de relief, d'un point de la surface vers le point de vue d'observation, pour déterminer ainsi si un point de la surface est occulté. Le cas échéant, les propriétés de réflectance du point courant sont remplacées par celles du point occultant. La Figure 1.10 illustre les résultats de cette méthode pour simuler le relief des pavés. Le même principe peut être appliqué pour approximer les effets d'ombrage provoqués par la méso-structure. Le lancé de rayon est réalisé d'un point de la surface vers la ou les sources lumineuses pour détecter si un point de la surface est dans une zone ombrée ou non. Ce type de méthode est visuellement réaliste pour des surfaces d'élévation simples mais nécessite des opérations coûteuses. De plus, l'unicité du résultat n'est pas garantie car la recherche des points occultants est non-linéaire mais, pour une question de performance, est réalisée ici par des méthodes linéaires.

1.2 Modèles de réflectance

Une surface réfléchit tout ou partie de l'énergie lumineuse à laquelle elle est soumise. La quantité d'énergie lumineuse réfléchie dépend de la direction d'incidence de la lumière et du point de vue d'observation, respectivement $\vec{\omega}_i$ et $\vec{\omega}_o$ sur la Figure 1.5. En fonction du matériau considéré (bois, métal, tissus,

etc), les propriétés de réflexion sont multiples et un modèle de réflectance est associé à un matériau en particulier ou une classe de matériaux aux propriétés physiques proches. Dans cette section, les différents modèles de réflectance de l'état de l'art sont présentés à travers une synthèse organisée en fonction des caractéristiques des matériaux auxquels chaque modèle peut s'adapter.

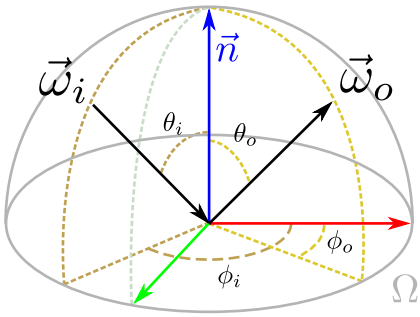


FIGURE 1.5: La direction lumineuse incidente $\vec{\omega}_i = (\theta_i, \phi_i)$ et le point de vue d'observation $\vec{\omega}_o = (\theta_o, \phi_o)$ représentés sur l'hémisphère Ω dans un repère local autour de la normale $\vec{n}$.

1.2.1 Matériaux uniformes

Considérons l'interaction entre un rayon lumineux et une surface tels que décrit sur la Figure 1.1 en simplifiant le système de la manière suivante :

- $t_i = t_r$, le temps de propagation de la lumière est négligeable,
- $x_i = x_r$, le trajet de la lumière est infime,
- $\lambda_i = \lambda_r$, l'interaction ne modifie pas la longueur d'onde.

Alors, la réflexion de la lumière est un procédé à deux paramètres, $\vec{\omega}_i$ et $\vec{\omega}_r$, liés deux à deux par la relation :

$$\vec{\omega}_r = \vec{\omega}_i - 2 * < \vec{n}, \vec{\omega}_i > * \vec{n}$$

La réflectance d'un matériau est évaluée pour un point de vue d'observation $\vec{\omega}_o$ et la « *Bidirectional Reflectance Distribution Function* » [Nic70] est une fonction modélisant la réflectance suivant un point de vue et une direction incidente de la lumière. On distingue deux caractéristiques principales pour un matériau qui va influencer la manière de définir la *BRDF*. Pour les matériaux isotropes, le calcul de la réflectance par une *BRDF* du type : telle que :

$$\rho\left(\theta_o, \phi_o, \theta_i, \phi_i - \phi_o\right)$$

avec $\vec{\omega}_o = (\theta_o, \phi_o)$, la direction du point de vue, et $\vec{\omega}_i = (\theta_i, \phi_i)$, la direction du rayon lumineux incident.

Cette propriété possède l'avantage de réduire le nombre de dimensions de la *BRDF* (trois dimensions seulement) qui devient invariante par la rotation autour de la normale. Dans le cas général, anisotrope, la *BRDF* est une fonction de quatre dimensions généralement définie comme :

$$\rho\left(\vec{\omega}_o, \vec{\omega}_i\right) = \rho_d\left(\vec{\omega}_o, \vec{\omega}_i\right) + \rho_s\left(\vec{\omega}_o, \vec{\omega}_i\right)$$

où les termes ρ_d et ρ_s correspondent respectivement à la composante diffuse et à la composante spéculaire qui sont deux comportements directionnels influents sur la réflectance de manière différente. Comme illustré sur la Figure 1.6, lorsque la distribution de l'énergie réfléchie par la surface est uniforme dans l'espace des directions, la réflexion est *lambertienne* (i.e. ρ_d). A l'inverse, lorsque l'énergie réfléchie est orientée principalement autour du vecteur de réflexion $\vec{\omega}_r$, on parle alors de réflexion spéculaire (i.e. ρ_s). Une surface est rarement complètement *lambertienne* ou spéculaire pure, on parle alors de surface *glossy* ; c'est pourquoi la *BRDF* prend en compte ces deux composantes et la répartition entre réflexion diffuse et spéculaire dépend des propriétés du matériau.

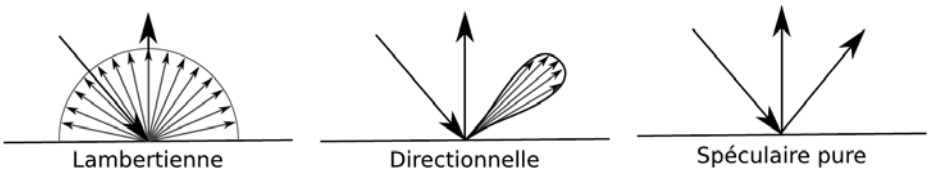


FIGURE 1.6: Différents types de réflexion. La réflexion *Lambertienne* est omnidirectionnelle. La réflexion spéculaire, directionnelle, est orientée autour du vecteur de réflexion. Une surface spéculaire pure réfléchit la totalité de l'énergie lumineuse suivant le vecteur de réflexion.

La *BRDF* est régie par des règles physiques importantes à travers le principe de *réciprocité* et la *conservation de l'énergie* [Nic70]. La *réciprocité* est définie telle que la réflectance du matériau est indépendante du sens du flux lumineux : $\rho\left(\vec{\omega}_o, \vec{\omega}_i\right) = \rho\left(\vec{\omega}_i, \vec{\omega}_o\right)$. La *conservation de l'énergie* décrit le comportement selon lequel l'énergie réfléchie par la surface ne peut excéder l'énergie reçue par celle-ci. Un aperçu des méthodes de l'état de l'art est présenté dans la section 2.2.1.

Tous les matériaux uniformes ne peuvent pas être représentés par une *BRDF*. En effet, cette fonction modélise uniquement les effets de réflexion d'un rayon lumineux à la surface d'un objet mais pas les propriétés d'absorption qui implique une diffusion de la lumière sous la surface de l'objet. Les matériaux translucides (lait, peau, marbre, *etc*) possèdent cette propriété et la modélisation de ce type de matériau implique indubitablement un ajout de paramètres. Pour cela, Jensen *et al.* [JMLH01] ont proposé la « *Bidirectional*



(a) *BRDF* (b) *BSSRDF*

FIGURE 1.7: Rendu de la peau à l'aide (a) d'une *BRDF* et (b) d'une *BSSRDF* [JMLH01].

Subsurface Scattering Reflectance Distribution Function », fonction de huit dimensions définie telle que :

$$\rho\left(x_o, \vec{\omega}_o, x_i, \vec{\omega}_i\right)$$

avec x_i et x_o qui sont, respectivement, le point d'entrée du rayon lumineux et le point sur lequel on cherche à calculer la réflectance. La *BRDF* est donc, en fait, un cas particulier de la *BSSRDF* pour des matériaux opaques. Une comparaison est illustrée sur la Figure 1.7 où l'on peut voir l'apport de la translucidité sur un matériau comme la peau.

1.2.2 Matériaux non-uniformes

L'utilisation de textures pour le stockage de coefficients est un un procédé souvent utilisé en informatique graphique. Comme vue précédemment, le *Bump Mapping*, bien que définit analytiquement, encode généralement les normales dans une texture et le *Relief Mapping* stocke les variations de hauteur de la méso-structure aussi dans une texture. Les surfaces hétérogènes n'ont pas, par définition, un modèle de réflectance unique sur l'ensemble de leur surface. Pour ce type de matériaux, les coefficients d'une *BRDF* varient spatialement et sont stockés dans une texture. Un exemple de cette méthode, la « *Spatially-Varying Bidirectional Reflectance Distribution Function* » (*SVBRDF*), est illustrée sur la Figure 1.8 avec deux types de matériaux non-uniformes. Ce modèle possède donc deux dimensions supplémentaires par rapport à la *BRDF* et est décrit comme :

$$\rho\left(x, \vec{\omega}_o, \vec{\omega}_i\right)$$

où x définit le point de la surface, en coordonnées de texture 2D, où la réflectance est évaluée.



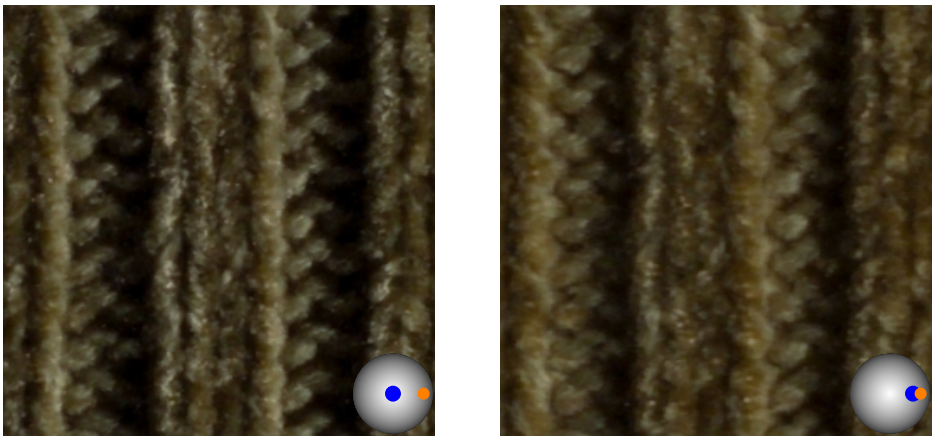
(a) (b)

FIGURE 1.8: Exemples de matériaux non-uniformes : (a) *SVBRDF* issue de [LBAD+06] (b) *BTF* provenant de [BTFa].

Les *SVBRDFs*, généralement issues de l'acquisition, sont représentées dans la littérature de deux manières différentes. Une première catégorie de méthodes [McA02, MLH02] considère cette représentation comme un ensemble de *BRDF*. Ainsi, la surface est discrétisée de manière à ce qu'un point de la surface corresponde à un pixel de la texture où ses propriétés de réflectance, paramètres du modèle de *BRDF* utilisé, sont stockées. Une autre catégorie de méthodes factorise les données acquises de réflectance à l'aide de décompositions matricielles [LRR04, LBAD+06, WWHL07]. Les données ainsi factorisées sont stockées dans une ou plusieurs textures. Notons que cette catégorie de méthode permet aussi l'édition des matériaux en projetant les données dans une base de *BRDFs* éditables à l'aide de factorisations matricielles non-négatives.

Bien que permettant de représenter un grand nombre de matériaux, les *SVBRDFs* souffrent d'une limitation importante : il est impossible de représenter une surface ayant une méso-structure complexe. En effet, dès qu'une surface est considérée comme une surface d'élévation, de nombreux effets interviennent dans le calcul de la réflectance : inter-réflexions, ombrages et occultations (Figure 1.9). En connaissant la géométrie de la surface, un calcul de visibilité peut être effectué pour modéliser en partie ces effets. Cette approche est toutefois coûteuse car elle nécessite à chaque rafraîchissement d'images durant le rendu le calcul de la surface visible, des ombres et des inter-réflexions.

Pour représenter un matériau avec une méso-structure complexe, Dana *et al.* [DNGK97] ont proposé les « *Bidirectional Texture Functions* » (*BTFs*). Issue d'un procédé d'acquisition, une *BTF* est un ensemble de textures directionnelles où une texture $T_{(\vec{\omega}_o,\vec{\omega}_i)}$ représente le matériau pour un point de



(a) (b)

FIGURE 1.9: Textures issues d'une *BTF* [BTFa] avec (a) $\vec{\omega}_i = (75;0)$ et $\vec{\omega}_o = (0;0)$ (b) $\vec{\omega}_i = (75;0)$ et $\vec{\omega}_o = (60;0)$. La direction rasante de la lumière fait apparaître des effets d'ombrage sur le matériau. Entre les deux points de vue, on constate des effets de recouvrement et découvrement de surface dus à la parallaxe.

vue $\vec{\omega}_o$ et une direction de la lumière $\vec{\omega}_i$:

$$BTF\left(x, \vec{\omega}_o, \vec{\omega}_i\right) = T_{(\vec{\omega}_o, \vec{\omega}_i)}(x)$$

Ici, les dimensions spatiales (i.e. variable x) n'ont plus tout à fait la même signification que dans les méthodes précédentes. Dans le cas d'une *SVBRDF*, x est un point de la surface exprimé de manière discrète, normalisé, et stocké dans une texture. Pour une *BTF*, afin d'y inclure les effets de parallaxe, chaque texture directionnelle est *rectifiée* telle que la normale à la texture soit alignée avec la direction représentant le sommet de l'hémisphère. Ainsi, x ne désigne plus directement un point de la surface mais la projection des points visibles de la surface pour un point de vue $\vec{\omega}_o$ donné. Un exemple est donné sur la Figure 1.10 où l'on peut observer la non-bijectivité entre les coordonnées de texture et les coordonnées surfaciques suivant les points de vue.

L'avantage d'une telle représentation est le nombre important de phénomènes physiques qu'elle peut représenter : occultations, ombrage, inter-réflexions, diffusion sous-surfacique, *etc*. Cependant, bien que représentant une alternative intéressante aux méthodes précédemment exposées, les *BTFs* souffrent d'un inconvénient majeur : la taille des données car une texture est nécessaire par échantillon angulaire. Dans ce manuscrit, pour définir les caractéristiques liées aux *BTFs*, les notations suivantes seront utilisées :

- $\mathbb{W}_o$, ensemble des directions du point de vue de la *BTF* ;
- η_o, nombre de points de vue inclus dans $\mathbb{W}_o$;

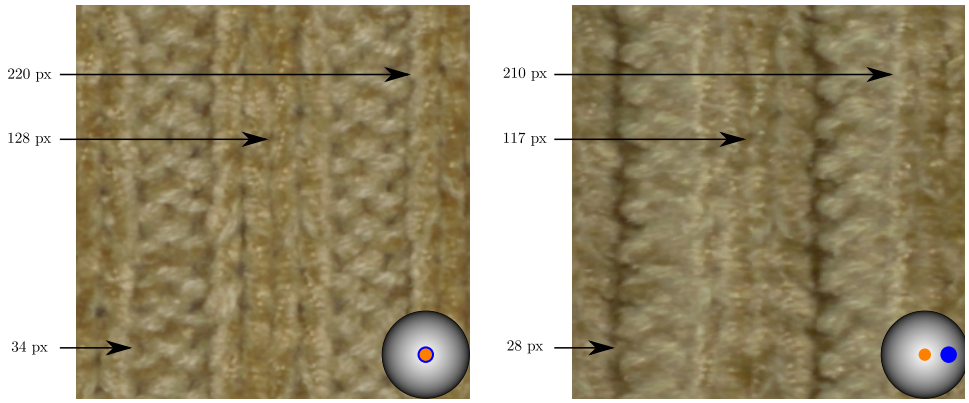


FIGURE 1.10: Mise en évidence des effets de parallaxe sur une *BTF*. Les deux textures représentent le même matériau avec $\vec{\omega}_o = (0;0)$, à gauche, et $\vec{\omega}_o = (45;0)$, à droite, suivant une lumière incidente normale, $\vec{\omega}_i = (0;0)$. Pour les mêmes points à la surface du matériau, les coordonnées de texture sont différentes entre l'image de gauche et de droite. Cette observation correspond aux effets de parallaxe.

- $\mathbb{W}_i$, ensemble des directions de la lumière de la *BTF* ;
- η_i, nombre de positions de la lumière dans $\mathbb{W}_i$;
- $\mathbb{T}$, ensemble des pixels présents dans un échantillon d'une *BTF* ;
- η_x, la résolution spatiale ou nombre de pixels dans $\mathbb{T}$.

Le nombre de coefficients pour une *BTF* est donc $\eta_o \times \eta_i \times \eta_x$ ce qui représente une taille de plusieurs giga-octets pour la plupart des matériaux disponibles publiquement. Ces problématiques seront abordées dans le chapitre 2 avec une mise en évidence des principaux axes de recherche présents dans la littérature sur les *BTFs*.

Chapitre 2

Bidirectional Texture Functions



(a) (b)

Figure 2.1: Comparaison entre un objet texturé avec (a) une texture unique et (b) une *BTF* issue de l'université de Bonn [BTFa].

Le réalisme d'une scène en images de synthèse est guidé par le mimétisme entre un objet virtuel et son homologue réel, tant au niveau géométrique que de l'apparence. Alors que les systèmes d'acquisition 3D sont devenus fréquents pour capturer la géométrie d'un objet, l'acquisition de l'apparence est encore un thème de recherche à part entière et reste difficilement exportable vers l'industrie. Certaines recherches portent sur l'acquisition d'un type de matériau en particulier [MPBM03] mais la modélisation de surface réaliste à l'aide de ces matériaux reste encore laborieuse. Cette difficulté est encore plus grande si l'on souhaite reproduire la méso-structure d'un matériau complexe hétérogène. Néanmoins, de nombreuses méthodes

hybrides ont été développées pour représenter géométriquement la méso-structure : des combinaisons entre les méthodes classiques de modélisation géométrique et des techniques de plaquage de texture pour représenter la méso-géométrie. Cependant, la multitude de phénomènes physiques engendrés par l'ajout d'une géométrie fine est rarement prise en compte et, le cas échéant, s'avère souvent coûteuse en temps de calcul (voir Chapitre 1).

Se basant sur ce paradigme, les « *Bidirectional Texture Functions* » [DNGK97] ont l'avantage de regrouper en une seule texture l'aspect géométrique de la méso-structure et l'ensemble des effets lumineux associés (voir Figure 2.1). Les *BTFs* sont des fonctions à six dimensions qui sont décrites par un ensemble de textures, représentant les dimensions spatiales, pour toutes les directions visibles du point de vue et de la lumière. Issues de l'acquisition, les *BTFs* requièrent un dispositif de capture coûteux qui nécessite de nombreux post-traitements pour être utilisables dans un processus de rendu.

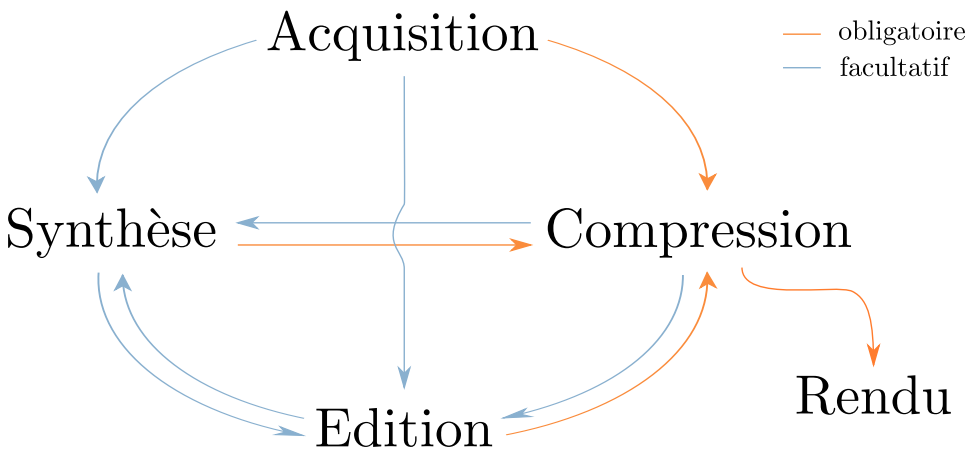


FIGURE 2.2: Arbre des possibles [Wer] du dispositif d'acquisition au rendu pour une *BTF*. On dénombre cinq domaines de recherche principaux pour les *BTFs* : l'acquisition, la compression, l'édition, la synthèse et le rendu. L'acquisition est bien entendu le point de départ pour générer une *BTF*. Alors que certains domaines sont facultatifs, la compression de *BTF* est le domaine le plus actif qui de plus est central et indispensable pour visualiser une *BTF*.

La Figure 2.2 présente les domaines de recherche associés aux *BTFs*. Le premier élément est l'acquisition de *BTFs* nécessaire pour numériser un matériau réel. Cette étape permet de générer un ensemble de textures directionnelles qui composent une *BTF*. Plusieurs processus d'acquisition de matériaux réels ont été proposés dans la littérature mais aussi des méthodes de génération de *BTFs* par pré-calcul des effets induits par la méso-géométrie (voir Section 2.1). Une deuxième étape indispensable pour utiliser une *BTF*

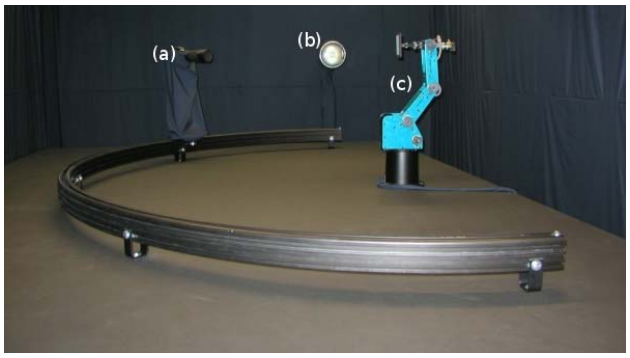
est l'étape de compression. En effet, la taille d'une *BTF* post-acquisition est de l'ordre du giga-octet et, bien que la capacité mémoire du matériel graphique soit sans cesse en expansion, il n'est pas envisageable d'occuper une telle place mémoire pour un seul matériau. Une phase de compression est donc indispensable pour minimiser l'espace des données en mémoire ou pour le stockage (voir Section 2.2). Notons que la phase de compression est d'autant plus importante qu'elle va guider les méthodes de rendu. Une exigence supplémentaire est donc requise sur ces méthodes pour permettre une visualisation en temps réel des données sur le matériel graphique. D'autres domaines en marge des problématiques principales des *BTFs* sont aussi représentés dans l'état de l'art : l'édition et la synthèse. L'édition consiste à ne pas considérer la *BTF* comme un matériau figé après acquisition mais comme un matériau éditable où certaines de ses propriétés peuvent être changées par un infographiste : couleur, ombrage, *etc.* La synthèse de *BTFs*, inspirée des techniques de traitement d'images, permet de générer un ensemble de textures bi-directionnelles de taille infinie par assemblage de motifs répétés, en prenant soin de minimiser les discontinuités (voir Section 2.3).

2.1 Acquisition

L'acquisition de *BTFs* est une étape onéreuse et complexe à mettre en place nécessitant d'une part, un dispositif d'acquisition et d'autre part, des post-traitements qui vont influencer la qualité d'une *BTF*. Les critères qui permettent de juger de l'efficacité d'un dispositif d'acquisition sont nombreux : temps d'acquisition, échantillonnage angulaire, résolution spatiale, source d'éclairement, acquisition HDR[1], déformation des échantillons rasants, *etc.* Un certain nombre de dispositifs a été présenté dans la littérature et les principales différences se situent sur le périphérique de capture, caméra ou appareil photographique numérique (APN), et le nombre de degrés de liberté laissé au capteur, à la lumière ou à l'échantillon.

Les pionniers dans ce domaine sont Dana *et al.* [DNGK97] dont les résultats ont été diffusés publiquement dans la base de données CUReT [Cur]. Basé sur des systèmes de mesure de *BRDFs* [CLM, War92], ce système reprend le dispositif de *gonioréflectomètre* en remplaçant le spectromètre, initialement suffisant pour la capture de *BRDFs*, par un capteur CCD (*Charge-Coupled Device*, ou dispositif à transfert de charge) pour capturer les variations spatiales. Bien que précurseurs dans ce domaine, ces travaux ont conduit à l'acquisition de 61 matériaux spatiaux isotropes dont l'échantillonnage angulaire trop éparse pour être utilisés dans un rendu haute qualité. Dû au positionnement manuel du capteur CCD, le nombre de configurations angulaires possibles est limité par le temps nécessaire au repositionnement du capteur. De plus, les données disponibles dans [Cur] sont brutes et requièrent

1. High Dynamic Range : images à forte dynamique.



(a) Capteur CCD mobile sur un rail.
(b) Source de lumière mobile sur un rail.
(c) Matériau posé sur un socle.

FIGURE 2.3: Dispositif d'acquisition de l'université de Bonn [BTFa].

donc des post-traitements afin d'être utilisables. Certains de ces matériaux ont par la suite été complétés par différentes échelles spatiales pour la classification de matériaux [HCFE04]. Une version améliorée du système CUReT est proposée par l'université de Bonn [SSK03] où la principale innovation réside dans la mobilité du capteur CCD, grâce à un rail automatisé, qui autorise un positionnement du capteur et de la source de lumière automatique pour permettre un échantillonnage angulaire dense pour un temps de traitement raisonnable. Ces données sont disponibles également sur des bases de données publiques [BTFa] et un exemple de ce dispositif est présenté sur la Figure 2.3. Le système est prévu pour une acquisition automatique d'une *BTF* tant au niveau des prises de vue sous différentes configurations d'éclairement qu'au niveau des post-traitements : détourage et alignement de la texture dans la direction normale du point de vue. Le temps de mesure avec un tel appareillage [BTFa] est environ de 14 heures pour un échantillonnage de $\eta_i = 81$ directions de la lumière et $\eta_o = 81$ directions de points de vue. Notons qu'ici, le facteur limitant est le temps de transfert du capteur CCD vers l'unité centrale de traitement. La résolution spatiale de chaque texture rectifiée est de $\eta_x = 256 \times 256$ et comporte 3 composantes colométriques RVB (Rouge Vert Bleu). Plusieurs travaux ont été proposés sur ce système pour la calibration du périphérique d'acquisition [Lys09] et sur la validation physique des données acquises [HNZ+05].

Koudelka *et al.* [KMBK03] ont proposé une autre approche sensiblement différente pour l'acquisition de *BTFs* dont l'élément le plus différentiable est l'utilisation d'une caméra numérique fixe comme périphérique d'acquisition. La source de lumière est dirigée par un bras robotisé alors que l'échantillon est placé sur un socle possédant 2 degrés de liberté (élévation et azimut) induit par le choix d'un récepteur fixe. L'utilisation d'une caméra vidéo au lieu d'un simple capteur CCD améliore le temps d'acquisition (environ 10 heures) mais limite la résolution de chaque texture directionnelle ($\eta_x = 192 \times 192$ texels). Néanmoins, ce système permet un échantillonnage plus dense

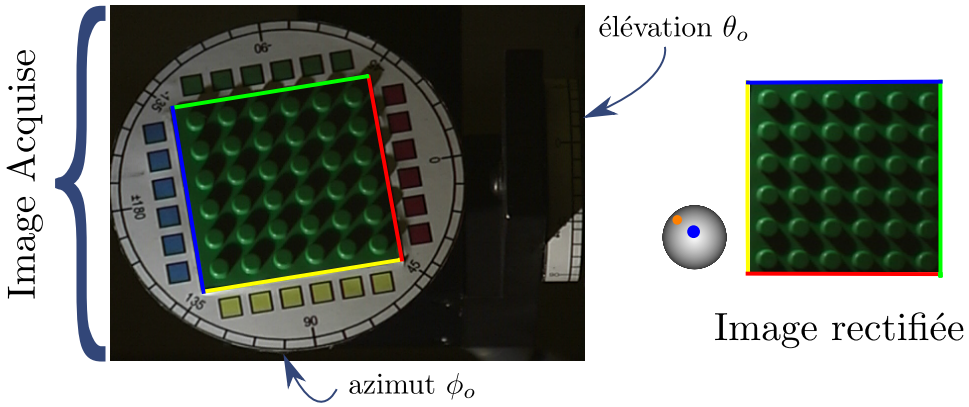


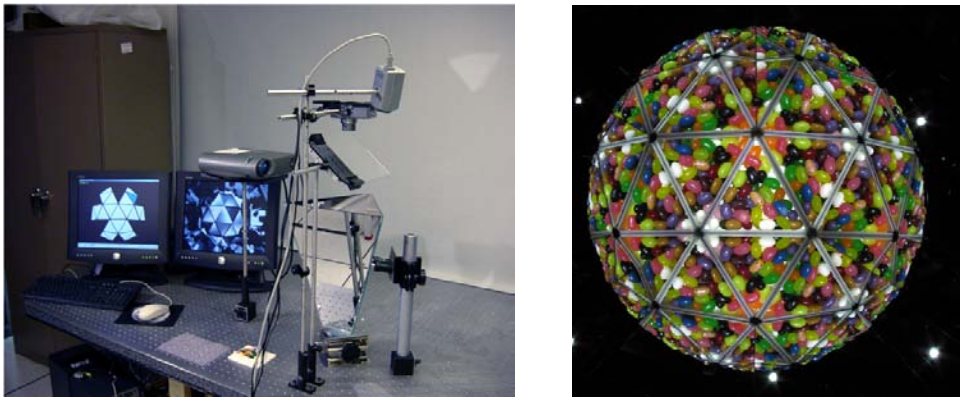
FIGURE 2.4: Résultat d'acquisition présenté dans [KMBK03]. Le matériau est placé sur un socle possédant deux degrés de liberté pour l'élévation et l'azimut. La lumière est placée sur un arc mobile et le dispositif de capture est fixe. La rectification de l'image acquise consiste à segmenter et projeter la texture dans le plan formé par la direction de vue normale.

de l'espace angulaire par une acquisition de $\eta_o = 90$ points de vue pour $\eta_i = 120$ directions de la lumière. Un exemple résultant de ce processus d'acquisition est présenté sur la Figure 2.4 sur laquelle on peut voir l'image issue de l'acquisition et l'image rectifiée qui composera la *BTF*.

Une idée originale est proposée par Han et Perlin dans [HP03, BR04] dont le principe consiste à utiliser un jeu de miroirs (i.e. un kaléidoscope) pour capturer plusieurs points de vue (resp. directions de lumière) en une seule prise. Le système est illustré sur la Figure 2.5 où l'on peut voir le dispositif de capture et une image acquise. Ici, seules les directions d'observation sont parallélisées et la source lumineuse, un vidéo projecteur, est partagée entre tous les points de vue en utilisant un *beam splitter*[2]. Les avantages de ce système sont d'une part, le temps d'acquisition et d'autre part, la précision de l'acquisition dans la mesure où toutes les composantes sont fixes. Cependant, de nombreux inconvénients sont à souligner. Premièrement, l'échantillonnage angulaire et la résolution spatiale sont liés : un échantillonnage important de l'espace des directions entraînera une résolution spatiale faible et inversement. Ensuite, la qualité de l'acquisition dépend de la qualité de réflexion des miroirs utilisés, dont aucun ne possède une réflexion parfaite. Bien que limitée, cette solution d'acquisition a le mérite d'être novatrice et laisse envisager des perspectives intéressantes pour de nouvelles méthodes d'acquisition à base de systèmes optiques complexes.

Dans un soucis de rapidité d'acquisition et de mobilité du système, Malzbender *et al.* [MGW01] ont proposé un dôme d'acquisition mobile (voir Figure

2. Traduction littérale : séparateur de faisceaux.

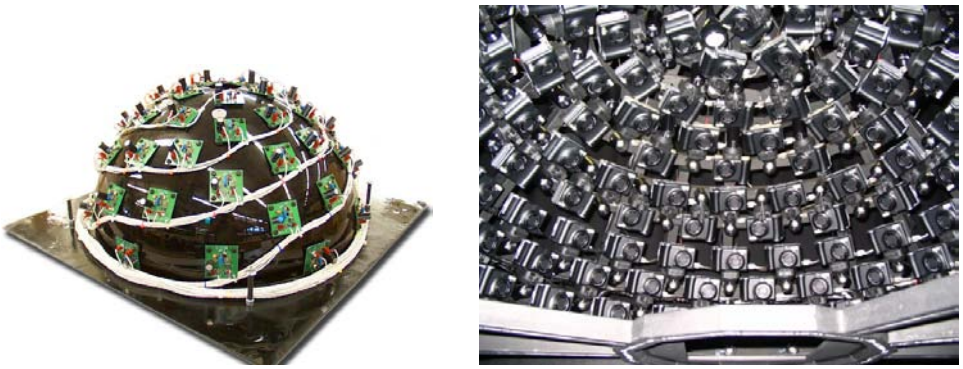


(a) (b)

FIGURE 2.5: Dispositif d'acquisition utilisant un kaléidoscope [HP03] pour capturer en une seule fois plusieurs points de vue.

2.6 (a)) autour du projet « Polynomial Texture Map » (PTM). Le dispositif est composé de $\eta_i = 50$ sources lumineuses équi-réparties sur un dôme et un capteur CCD placé au sommet. L'avantage de cette méthode réside dans la mobilité du dispositif qui permet de capturer de nombreux matériaux du quotidien rapidement. Cependant, un unique point de vue limite l'utilisation des matériaux spatiaux capturés dans une application de rendu réaliste. Respectant la même approche, Müller *et al.* [MBK05] ont proposé le même type de dôme mais en multipliant le nombre de capteurs CCD (voir Figure 2.6 (b)). Ici, les sources lumineuses et les capteurs CCD sont regroupés en un seul périphérique via un APN équipé de flashs. 151 APN sont ainsi équi-répartis sur le dôme d'acquisition et reliés à une unité centrale de contrôle par une connectique USB. Grâce à cette parallélisation massive de l'acquisition, une *BTF* est acquise en 40 minutes. Initialement prévu pour l'acquisition de la géométrie et de l'apparence d'un objet, nous n'avons pas d'information sur la résolution spatiale des images rectifiées. Les inconvénients d'un tel dispositif sont d'une part le coût (151 APN !) et d'autre part, l'échantillonnage angulaire fixe qui ne permet pas de capturer plus finement certaines régions angulaires importantes d'un matériau spécifique.

Les matériaux surfaciques obtenus grâce aux systèmes d'acquisition décrits précédemment permettent, dans l'ensemble, d'agrémenter le réalisme des objets virtuels facilement par des techniques de plaquages de texture classiques lors du rendu. Néanmoins, de nombreux problèmes dans le processus d'acquisition restent encore ouverts. Le point critique dans l'acquisition est le lien entre le dispositif d'acquisition et les post-traitements. En effet, pour l'ensemble de procédés énoncés dans cette section, toute image acquise doit être rectifiée. Cette rectification consiste d'une part, à segmenter l'image capturée suivant des marqueurs disposés lors de l'acquisition pour extraire les

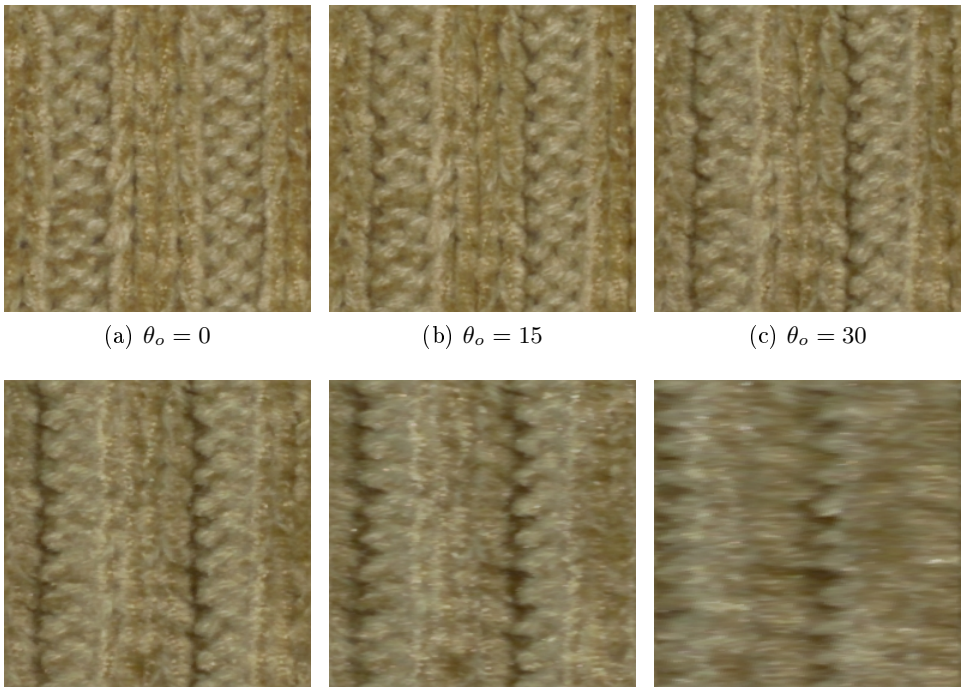


(a) (b)

FIGURE 2.6: Dômes d'acquisition de *BTF* issues (a) du projet PTM [MGW01] et (b) de l'université de Bonn [MBK05].

régions contenant le matériau uniquement et, d'autre part, corriger la perspective induite par le changement de point de vue afin de projeter l'ensemble des textures directionnelles dans le même plan décrit par les coordonnées polaires $(0, 0)$. Ce procédé produit des textures telles qu'illustrées sur la Figure 2.7 pour différents angles d'élévation. L'information sur les angles rasants (i.e. $\theta_o > 60$) n'est pas pertinente pour plusieurs raisons. Tout d'abord, il n'est pas logique d'attribuer la même résolution spatiale, ou quantité d'information, pour les différents angles d'élévation : la résolution spatiale est inversement proportionnelle à l'angle d'élévation θ_o. Ensuite, la correction de perspectives nécessaire à la projection de la texture dans le plan normal induit de plus fortes erreurs pour des angles de forte élévation. De ce fait, la correspondance entre les point visibles à la surface du matériau et les coordonnées de texture correspondantes, non bijective suivant le point de vue pour encapsuler les effets de parallaxes, n'est plus respectée. Un système d'acquisition plus complexe au niveau optique pourrait évincer cette limitation en intégrant directement toute ou partie de la phase de rectification au sein du dispositif d'acquisition par redirection du signal spatial dans une configuration adéquate pour le périphérique d'acquisition : taille de l'échantillon constante quelque-soit l'angle d'élévation. Bien que cette observation ne soit pas perceptuellement visible lors du rendu de *BTFs*, elle limite les algorithmes de reconstruction d'informations géométriques de la surface du matériau (normales ou hauteur). Notons que Wang *et al.* [WWS+05a] ont proposé un système d'acquisition « classique » agrémenté d'un laser pour mesurer et visualiser la silhouette sur les *BTFs*.

Nous nous sommes intéressés ici uniquement à l'acquisition de matériaux réels mais d'autres systèmes de génération de *BTFs* synthétiques ont été développés [DLHS01, SBLD03, VT04, KBD07, Had09] qui sont, d'un point de vue personnel, pas assez réalistes pour être utilisables : phénomènes



(a) $\theta_o = 0$ (b) $\theta_o = 15$ (c) $\theta_o = 30$

(d) $\theta_o = 45$ (e) $\theta_o = 60$ (f) $\theta_o = 75$

FIGURE 2.7: Textures issues d'une *BTF* (Pulli [BTFa]) pour des angles d'élévation θ_o croissants et de même azimut (i.e. 0 degré) pour la même direction de la lumière $\omega_i = (0, 0)$.

physiques limités ou méso-structure trop artificielle. Dans ce manuscrit, nous utiliserons uniquement les *BTFs* issues de la « BTF Database Bonn » [BTFa] et de la « Volumetric Surface Texture Database » [KMBK03] qui sont les *BTFs* de qualité supérieure parmi celles disponibles publiquement.

2.2 Compression

La compression est un élément indispensable en post-acquisition qui consiste à minimiser l'espace de stockage requis pour une *BTF* tout en limitant les erreurs d'approximation. La taille d'une *BTF* brute dépend de l'échantillonnage utilisé lors de l'acquisition dans l'espace angulaire et de la résolution spatiale capturée pour chaque texture. Dans le cas des *BTFs* de [BTFa], l'échantillonnage angulaire est de $\eta_o = 81$ directions d'observation pour $\eta_i = 81$ directions de la lumière et une résolution spatiale de $\eta_x = 256 \times 256$ pixels au format RVB. Le nombre de coefficients depuis les données brutes est donc de :

$$\eta_x \times \eta_o \times \eta_i \times 3 = 256 \times 256 \times 81 \times 81 \times 3 = 1\,289\,945\,088$$

En considérant chaque coefficient comme un réel positif ($\in \mathbb{R}^+$) codé sur 16 bits (*half-float*), la taille d'une *BTF* brute est de $2,4$ Go. Manipuler une telle quantité de mémoire, même sur les machines actuelles, n'est pas efficace. De plus, dans une scène virtuelle, tous les éléments inclus dans la scène partagent les mêmes ressources et la mémoire limitée du matériel graphique implique un taille restreinte pour un matériau, de l'ordre de quelques dizaines de méga-octets. Le ratio de compression nécessaire pour des applications typées rendu est donc de l'ordre de 1 : 100. Pour obtenir de tels résultats, on distingue deux approches pour la réduction des données :

- « *lossless* », compression sans perte ;
- « *lossy* », compression avec erreur.

Dans la première approche, les données d'origine sont retrouvées à l'identique depuis les données compressées. A l'inverse, lors d'une compression *lossy*, la reconstruction des données compressées est proche des données d'origine à une erreur d'approximation près. Les coefficients issus d'une *BTF* sont fortement corrélés à la fois dans le domaine angulaire mais aussi dans le domaine spatial. Exploiter au maximum la corrélation du jeu de données n'est souvent pas suffisant pour obtenir des ratios de compression élevés : le type de compression retenu pour les *BTFs* est donc *lossy*.

Les critères d'évaluation des méthodes de compression sont, en premier lieu, la ratio de compression par rapport à l'erreur introduite mais aussi, les facultés d'une représentation compacte à s'adapter au matériel graphique. En effet, l'intérêt principal d'une représentation légère de *BTF* est de définir le matériau d'un objet lors du rendu à l'aide de la structure de données compacte. La reconstruction de la *BTF* est réalisée via les processeurs graphiques programmables, les *shaders*, et la complexité des algorithmes de reconstruction est donc un critère important dans une application en temps réel.

Comme illustré sur la Figure 2.8, une *BTF* peut être définie de deux manières différentes :

- un ensemble de textures directionnelles ;
- un ensemble de *ABRDFs*.

Une *ABRDF* [WHON97], pour « *Apparent Bidirectional Reflectance Distribution Function* », est l'équivalent d'une *BRDF* anisotrope mais elle n'est pas soumise aux propriétés physiques de réciprocité et conservation de l'énergie (voir Chapitre 1). De plus, une *ABRDF* inclut le terme d'atténuation entre la normale et la direction de la lumière, le facteur cosinus $< \vec{n}, \vec{\omega_i} >$, et de nombreux effets provoqués par la méso-structure, occultation, ombrage et inter-réflexions.

Une première catégorie de méthodes considère une *BTF* comme un ensemble de *ABRDFs* (2.2.1). Ainsi, chaque pixel est approximé par un modèle paramétrique de *BRDF*. D'un autre coté, une autre méthode pour compresser une *BTF* est d'utiliser des techniques de traitement du signal qui utilisent une décomposition dans une base linéaire pour réduire la dimensionnalité (2.2.2). Enfin, une approche consiste à exploiter la corrélation des

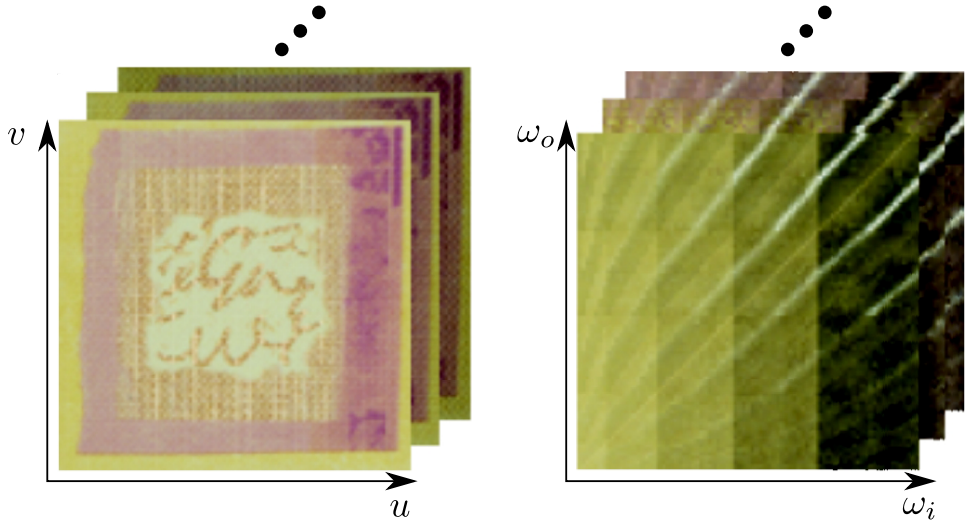


FIGURE 2.8: Deux configurations pour une *BTF* : un ensemble de textures (à gauche) ou un ensemble de *ABRDFs* (à droite).

données angulaires pour substituer un échantillon par son voisin de plus grande vraisemblance (2.2.3). D'autres méthodes adoptent une approche probabiliste pour compresser les données. Cependant, il n'existe pas de fonctionnelle permettant de juger de la qualité de cette dernière représentation, hormis une metrique définie par des tests utilisateurs ! De ce fait, ces travaux seront abordés séparément dans la Section 2.3.

2.2.1 Compression basée approximation

2.2.1.1 Principe

Les méthodes de compression de *BTF* qui approximent les *ABRDFs* par pixel sont inspirées des travaux de McAllister [McA02] sur les *SVBRDFs*. Le principe de ses travaux consiste à évaluer chaque pixel d'un matériau surfacique issu de l'acquisition par un modèle paramétrique de *BRDF*. Les nombreuses méthodes de la littérature peuvent être divisées en deux catégories : les méthodes basées sur une formulation analytique de la *BRDF* et celles basées sur des méthodes numériques pour approximer des *BRDF* mesurées par un processus d'acquisition.

Les méthodes de la première catégorie comportent différentes formulations mathématiques pour représenter les phénomènes de réflectance de manière empirique. Le principe de ces méthodes consiste à formaliser un modèle de réflectance par un ensemble de paramètres contrôlables pour reproduire le comportement d'un matériau réel. Des modèles ad-hoc efficaces [Pho75, Bli77] ont été développés mais aussi des modèles plus réalistes comportant des effets visuels complexes [HTSG91, Sta99] dépendant

du nombre de phénomènes pris en compte. Les modèles les plus populaires [CT82, War92, Sch94b, APS00, EBJ+06, AP07] reposent sur une distribution de *micro-facettes* [TS67] qui définit une surface réfléchissante comme un ensemble de minuscules facettes orientées aléatoirement, chacune agissant comme un miroir de Fresnel. La plupart des modèles existants est référencée dans des revues de l'état de l'art [Sch94a, SMSA01, DR08].

D'un autre coté, le but des méthodes basées approximation est de reproduire les données d'une *BRDF* mesurée expérimentalement sur un matériau réel. Le principe consiste à projeter les données mesurées dans une base de fonctions en utilisant des méthodes d'algèbre linéaire classiques telles que la décomposition en valeurs singulières. Une première famille de publications s'appuie sur la projection dans des bases sphériques telles que les harmoniques sphériques [CMS87, WAT92], les polynômes de Zernike [KDS96], les ondelettes sphériques [SS95] ou des bases de fonctions radiales sphériques [ZERB05]. Une deuxième famille de méthodes [KM99, MAA01, SBLD03] s'appuie sur des méthodes de réduction de dimensionnalité et sera présentée dans la Section 2.2.2.

Les *BRDF*s issues de l'acquisition peuvent aussi être approximées par des modèles paramétriques. Certains modèles analytiques ont été spécifiquement développés pour ce type de méthode numérique d'approximation [War92, LFTG97]. Dans les travaux de McAllister [McA02], le modèle retenu est celui de Lafortune [LFTG97], connu pour ses propriétés d'adaptativité à plusieurs types de matériaux [NDM05]. Un lobe du modèle de Lafortune est défini comme :

$$\rho_l(\vec{\omega}_o, \vec{\omega}_i) = (\vec{\omega}_o^t . C . \vec{\omega}_i)^n$$

où C est une matrice de 3×3 qui, associée à l'exposant n, permet de définir la forme du lobe. Pour limiter le nombre de paramètres, la matrice C est réduite à une matrice diagonale ce qui limite la « forme » d'un lobe de Lafortune mais aussi les instabilités numériques. Une *ABRDF* est donc approximée par une somme de lobes de Lafortune et l'équation à résoudre est donc :

$$\begin{aligned} SVBRDF(x, \vec{\omega}_o, \vec{\omega}_i) &\approx \overline{SVBRDF}(x, \vec{\omega}_o, \vec{\omega}_i) \\ &\approx \alpha_{x,d} + \sum_j \alpha_{x,l_j} . \rho_{x,l_j}(\vec{\omega}_o, \vec{\omega}_i) \end{aligned} \tag{2.1}$$

où $\alpha_{x,*}$ représente l'albédo au point $x \in \eta_x$ et j fixe le nombre de lobes utilisés. Ce type d'équation n'est pas mathématiquement aisé à résoudre. En effet, les inconnues de l'équation à minimiser sont les albédos $\alpha_{x,*}$, la matrice C mais aussi l'exposant n qui implique l'utilisation de méthodes d'optimisation non-linéaires pour résoudre le système.

Dans le cas général, la méthode utilisée pour l'optimisation des paramètres est l'algorithme de Levenberg-Marquard (LM) [PFTV88]. Cette méthode est un processus itératif qui, à partir de paramètres d'initialisation p proches du

minimum global, permet de trouver un minimum local minimisant la fonctionnelle :

$$E(p) = \sum_i (f(i) - \overline{f}_p(i))^2$$

où f est la fonction d'origine et $\overline{f_p}$ est la fonction approximante avec les paramètres p suivant i valeurs. A chaque itération, on remplace les paramètres p par une nouvelle estimation $p+q$, telle que $E(p+q) \approx 0$, où q est la solution du système :

$$(J^t J)q = J^t D$$

où J est la matrice jacobienne telle que $J = \frac{\partial f_p}{\partial p}$ et D, le vecteur issu de la linéarisation du résiduel $(f(i) - \overline{f}_p(i))$ pour tous les échantillons i. Cette équation caractérise les méthodes dites de *descente de gradient*. La force de l'algorithme LM par rapport aux autres méthodes existantes est l'introduction d'un facteur d'*amortissement* positif λ :

$$(J^t J - \lambda I)q = J^t D$$

Le scalaire λ permet de fixer un pas plus ou moins grand à chaque itération pour se rapprocher plus ou moins rapidement de la solution.

De nombreux problèmes numériques découlent d'une telle méthode de minimisation. Premièrement, l'algorithme LM permet de trouver un minimum local, et non pas global, ce qui implique d'initialiser les paramètres avec des valeurs proches de la solution optimale. Or, lorsque le système est défini comme préconisé par McAllister [McA02], l'initialisation est complexe ! Deuxièmement, les temps de calcul sont relativement longs car il faut résoudre à chaque itération un système linéaire. Enfin, le nombre de paramètres à minimiser ne doit pas être trop élevé, ce qui est une conséquence directe des deux limitations évoquées précédemment. Cette observation fait que le nombre de lobes de Lafortune est restreint à, au plus, 3.

2.2.1.2 Méthodes

La factorisation d'*ABRDF* par un modèle paramétrique de *BRDF* n'est pas directe car, à l'heure actuelle, il n'existe pas de modèle de *BRDF* permettant de représenter, par lui même, tous les effets inclus dans une *ABRDF*. Ainsi, Daubert *et al.* [DLHS01] utilise le même type de modèle que celui présenté à l'équation 2.1 en y ajoutant un facteur multiplicatif pour modéliser les phénomènes induis par la méso-structure :

$$BTF(x, \vec{\omega}_o, \vec{\omega}_i) \approx T_x(\vec{\omega}_o).(\alpha_{x,d} + \sum_j \rho_{x,l_j}(\vec{\omega}_o, \vec{\omega}_i))$$

La majorité des effets méso-structurels étant localisée lors des changements de point de vue, le terme $T_x(\vec{\omega}_o)$ est défini par point de vue et par

pixel. La partie droite de l'équation approximant la *BRDF* est définie sur la luminance uniquement. Chaque élément de T est un vecteur de quatre composantes en RVBA où les trois coefficients colorimétriques RVB représentent la couleur du matériau et l'alpha (A) modélise les effets de la méso-structure en spécifiant l'opacité d'un point. Cette méthode fournit de bons résultats qualitatifs grâce à l'ajout du facteur multiplicatif T_x mais le coût quantitatif engendré limite son utilisation. Elle permet aussi de représenter des matériaux iridescents [3], phénomène fréquent sur des matériaux représentant des tissus maillés.

Dans un contexte où le point de vue est unique, Malzbender *et al.* [MGW01] ont proposé les « Polynomial Texture Maps » (PTMs). Cette méthode consiste, pour un point de vue et différentes directions de la lumière, à factoriser les variations de réflectance dus au changement de la position de la lumière par un polynôme bi-quadratique défini comme :

$$P(u, v, \Pi(\vec{\omega}_i)) = \sum_{i=0}^{2} \sum_{j=0}^{2} a_{i,j}(u, v).l_u^i.l_v^j$$

avec $\Pi(\vec{\omega}_i) = (l_u, l_v)$ projection orthographique de la direction de la lumière dans l'espace (u, v) de la texture, $a_{i,j}$ coefficients du polynôme par pixel. Cette approche a le mérite d'être très facile à mettre en place et d'obtenir des résultats satisfaisants si les propriétés de reflexion du matériau sont proche du lambertien.

Les PTMs ont par la suite été généralisés aux *BTFs* en approximant un polynôme bi-quadratique par point de vue. Considérer chaque point de vue indépendamment permet de ne pas être dépendant des effets de parallaxe et donc, de conserver une bijectivité entre les coordonnées de texture et les coordonnées de la surface. Cette approche est utilisée par Meseth *et al.* [MMK03a, MMK04a] en définissant une *BTF* comme un ensemble de points de vue discrets :

$$BTF\,(x, \vec{\omega}_o, \vec{\omega}_i) = \sum_j w_{\vec{\omega}_{o_j}} RF_{x,\vec{\omega}_{o_j}}(\vec{\omega}_i)$$

où les $w_{\vec{\omega}_{o_j}}$ sont les poids de pondération barycentrique pour l'interpolation des points de vue, et $RF_{x,\vec{\omega}_{o_j}}(\vec{\omega}_i)$, ou *reflectance field*, est la fonction de réflectance polynomiale définie par pixel x et par point de vue, similaire aux PTMs. L'avantage d'une telle méthode est que les surfaces à méso-structure complexe peuvent être représentées car la fonction polynomiale se comporte bien avec les effets d'ombrages. En revanche, les effets spéculaires ne sont pratiquement pas pris en compte. Pour parfaire ce comportement, une autre approche a également été proposée en utilisant des lobes de Lafortune à la place des polynômes. Cependant, les lobes de Lafortune seuls ne peuvent pas

3. Changement de la couleur du matériau suivant le point de vue d'observation.

prendre en compte les effets d'ombrage. De plus, le nombre de coefficients générés par cette méthode est encore trop important pour le rendu, 120 Mo pour l'approche polynomiale, et pourrait être reduit par des méthodes de décorrélation spatiale.

Suivant la même approche, Filip et Haindl [FH04] ont proposé un modèle basé sur une double interpolation (suivant les points de vue et les directions de la lumière) d'une extension polynomiale d'un lobe de Lafortune. L'extension polynomiale consiste, à partir d'une approximation avec un seul lobe, à changer la distribution des coefficients telles que les histogrammes cumulatifs entre les données d'origine et les données approximées soient identiques. Les résultats qualitatifs de cette méthode sont sensiblement les mêmes que ceux présentés dans [MMK03a] avec une taille de données plus importante. Ma *et al.* [MCC+04] définissent l'approximation d'une *BTF* comme une approximation grossière de la réflectance plus un résiduel qui peut être approximé à son tour par des fonctions simples. Ainsi, ils effectuent une première approximation par le modèle de Phong [Pho75] et le résiduel entre cette approximation et les données originales est représenté comme une composante spéculaire [Bli77] par pixel. Cette approche produit une approximation satisfaisante des données et permet une visualisation en temps-réel.

2.2.1.3 Discussion

Une première observation que l'on peut extraire de ces méthodes d'approximation est l'efficacité des modèles paramétriques de *BRDF* pour le rendu. En effet, les expressions arithmétiques d'un lobe de Lafortune ne nécessitent pas de gestions particulières des données sur le matériel graphique actuel et, en stockant les paramètres sous forme de texture, la visualisation de ces modèles est réalisée en temps réel. De plus, l'interpolation spatiale est gérée par la carte graphique, via le filtrage de textures, et l'espace des directions de la lumière est continu grâce au modèle paramétrique. Ainsi, dans le cas général, seule l'interpolation des points de vue est nécessaire.

Cependant, toutes ces approximations exploitent uniquement la corrélation dans l'espace angulaire mais jamais la cohérence spatiale, forte sur une *BTF*. Une des conséquence directe est que la taille des données reste très importante (environ 100 Mo). De plus, il n'y pas de niveaux de détail ni spatiaux ni angulaires. Dans l'espace des directions, cette restriction est due au choix du modèle de *BRDF*, Lafortune, alors qu'un modèle multi-résolution (ondelette sphérique ou harmonique sphérique) serait nécessaire pour une telle propriété. Néanmoins, l'ajout de plusieurs échelles spatiales est trivial et favoriserait d'une part, les niveaux de détail des textures pour le rendu et d'autre part, les temps de calculs de l'approximation. En effet, un des points limitant de ces méthodes est le temps de calcul. Utiliser des méthodes numériques non-linéaires est coûteux en temps et une approche multi-échelle est souvent bénéfique pour tendre vers un minimum plus rapidement. De

plus, l'optimisation des paramètres permet uniquement de trouver un minimum local ce qui implique une initialisation des paramètres très précise et donc une complexité temporelle accrue.

Enfin, l'utilisation de modèle de *BRDF* pour approximer une *ABRDF* restreint le type de matériaux surfaciques utilisable. En effet, lorsqu'une surface peut être considérée comme une surface d'élévation, les effets engendrés par la méso-géométrie sont nombreux. Le plus gênant dans cette approche est la gestion des ombres et notamment des ombres douces. Le comportement d'un lobe de Lafortune pour approximer les ombres est chaotique et donc, à proscrire pour des surfaces en 2.5D. Toutefois, l'approche polynomiale révèle un comportement sain pour la plupart des effets compris dans une *BTF*. Bien que limité dans son expressivité, les modèles polynomiaux sont très rapides à mettre en place, visuellement corrects (sauf matériau fortement spéculaire) et adaptés à un rendu temps réel sur matériel graphique.

2.2.2 Compression basée factorisation

2.2.2.1 Principe

Les méthodes de compression basées factorisation considèrent la *BTF* comme un signal multidimensionnel dont toutes ou parties des dimensions possèdent des informations redondantes ou inutiles qui peuvent être extraites du signal. La quantité de signal amputée au signal d'origine déterminera l'erreur d'approximation.

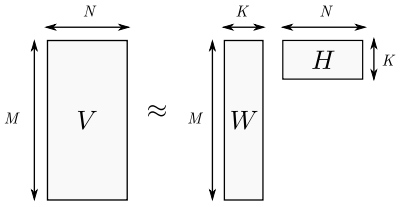


FIGURE 2.9: Factorisation matricielle. Une matrice V de taille $M \times N$ est factorisée par un produit de deux matrices $W * H$ de taille respective $M \times K$ et $K \times N$.

Le principe en est très simple : en linéarisant les données d'une *BTF* sous forme matricielle (matrice V de taille $M \times N$) on peut approximer cette matrice par une produit de deux matrices, W et H, de taille plus petite, respectivement $M \times K$ et $K \times N$. Toute la difficulté réside dans la construction de la base d'approximation H qui minimise les erreurs d'approximation. On retrouve dans la littérature deux types de méthodes pour obtenir une base : l'Analyse en Composante Principale (ACP) ou la *Non-negative Matrix Factorization* (NMF).

Le principe a été développé par Nishino *et al.* [NSI01] avec les *eigentextures*, dans un but de compresser des images d'un objet, prises sous dif-

férents points de vue, pour un rendu basé images. Dans cette approche, chaque texture est considérée comme un vecteur colonne de taille $M = \eta_x$. Les $N = \eta_o$ vecteurs colonnes (i.e. les textures) sont agencés les uns à coté des autres pour former une matrice V de taille $M \times N$. Après avoir centré la matrice V, le calcul des valeurs propres de la matrice de covariance V^tV est réalisé à l'aide d'une ACP. La matrice résultante H est une base orthogonale formée par les vecteurs propres de la matrice V^tV. Cette base minimise la variance du signal résiduel et forme une base linéaire optimale au sens des moindre carré. La matrice H est de taille $N \times N$. En ordonnant les vecteurs propres par ordre croissant de leur valeur propre correspondante, un sous-ensemble formé des K premiers vecteurs propres (i.e. les K premières lignes de la matrice H) est suffisant pour définir une base d'approximation, avec $K << N$. La matrice W est la résultante de la projection des valeurs de la matrice V dans la base formée par H.

2.2.2.2 Méthodes de factorisation

Factorisation d'*ABRDFs*

Les méthodes basées factorisation se catégorisent suivant l'agencement privilégié des données d'une *BTF* décidé dans chaque méthode. Ainsi, on retrouve l'approche par pixel où chaque *ABRDF* est compressée par factorisation matricielle individuellement. La formule générale de ces méthodes peut être généralisée comme :

$$ABRDF(\omega_i, \omega_o) \approx \prod_{j=1}^{J} \sum_{k=1}^{K_j} P_{j,k}(\pi_{j,1}(\omega_i, \omega_o)).Q_{j,k}(\pi_{j,2}(\omega_i, \omega_o)) \qquad (2.2)$$

où les $\pi(\omega_i, \omega_o)$ sont une projection 4D vers 2D de l'espace angulaire et P et Q, des fonctions 2D vers $\mathbb{R}$.

Les pionniers dans ce type de factorisation sont Kautz *et al.* [KM99] dont leur représentation est semblable à l'équation 2.2 avec $J = 1$. Une décomposition en valeur singulière (DVS) est utilisée ici pour approximer une *ABRDF* 4D en une somme de produit de deux fonctions 2D. Les paramètres P et Q peuvent ainsi être stockés sous forme de texture pour un rendu temps réel d'une *BTF*. Un compromis entre la qualité d'approximation et l'efficacité du rendu peut être trouvé en contrôlant le nombre de somme par *ABRDF*. Par la suite, McCool *et al.* [MAA01] ont proposé une factorisation homomorphique pour approximer chaque *ABRDF*. Leur représentation est de la forme de l'équation 2.2 avec $K_j = 1$ et $Q_{j,k} = 1$. Au lieu de considérer la factorisation comme une somme de produits, l'*ABRDF* est approximé par un nombre, contrôlable, de facteurs positifs. Suykens *et al.* [SBLD03] généralise ce type d'approche en utilisant une représentation correspondant à l'équation 2.2. Cette « *Chained Matrix Factorization* » est obtenue par une simple

séquence de factorisation, en utilisant la DVS par exemple, mais à chaque fois en utilisant une paramétrisation π différente. Les résultats améliorent les factorisations d'*ABRDF* précédentes et permet, en plus, d'abaisser la dynamique des coefficients pour une quantification plus efficace.

Factorisation par ACP

L'approche par analyse en composante principale initiée par Nishino *et al.* [NSI01] est la plus efficace pour obtenir une base d'approximation optimale au sens des moindres carrés. Cette méthode est utilisée régulièrement dans les méthodes de compression de *BTFs* et les différences entre les travaux existants portent sur le domaine sur lequel l'ACP est exécutée.

Ainsi, Koudelka *et al.* proposent dans [KMBK03] de factoriser dans la *BTF* dans son entier en utilisant l'ACP. Les échantillons sont organisés dans une matrice telle que les $\eta_i \times \eta_o$ textures directionnelles, chacune linéarisée sous forme de vecteur de taille η_x, forment les colonnes de la matrice V. Dans leur approche, Koudelka *et al.* utilisent des *BTFs* d'échantillonnage : $\eta_o = 90$ et $\eta_i = 120$. Ainsi, une fois centrée, la matrice V^tV est décomposée à l'aide d'une ACP pour former un ensemble de $\eta_i \times \eta_o = 10\,800$ vecteurs propres. Ces travaux ont déterminé que seulement 150 vecteurs propres sont suffisant pour représenter les différents effets d'une *BTF* efficacement. Bien que très convaincant en terme de compression, cette méthode ne permet évidemment pas une reconstruction en temps réel des données : $\eta_x \times 150$ additions et multiplications sont nécessaires pour reconstruire un pixel !

Plutôt que de considérer la factorisation de *BTFs* comme une matrice représentant la *BTF* complète à décomposer, Sattler *et al.* [SSK03] présentent une méthode où les échantillons sont regroupés par point de vue. Le principe est identique à la méthode précédente sauf qu'ici, les vecteurs propres de η_o matrices V de taille $\eta_x \times \eta_i$ sont calculés indépendamment. Cette méthode est plus facile à implémenter que l'approche *BTF* complète car la taille des matrices est raisonnable. D'autre part, la corrélation entre les textures est plus fortes que la corrélation globale d'une *BTF* ce qui permet de choisir un nombre de vecteurs propres petit pour la base d'approximation, 8 à 16 par point de vue suivant les matériaux. D'un point de vue compression, la taille des données est à peu près identique à la méthode *BTF* complète. Cependant, décomposer le problème global en sous problèmes permet une visualisation des données en temps interactif : au plus $\eta_x \times 16$ additions et multiplications sont désormais nécessaires à la reconstruction d'un pixel.

Pour décomposer le problème de factorisation de manière plus locale, Müller *et al.* [MMK03b, MMK04b] introduisent une factorisation par *clusters* en regroupant les échantillons de même vraisemblance. Le principe est semblable à la méthode de classification des *K-means* : partitionnement des échantillons en K groupes, où chaque échantillon est associé au groupe de moyenne la plus proche. Ici, la distance entre un échantillon et le centroid

d'un cluster est calculée dans l'espace propre et représente l'erreur d'approximation de la projection d'un échantillon dans cet espace propre. Les auteurs proposent deux modes de compression : (i) factorisation des textures ou (ii) factorisation des *ABRDFs*. La construction locale des matrices à décomposer est identique aux méthodes précédentes et chaque matrice regroupe uniquement les échantillons de même vraisemblance, V de taille $M \times N = \eta_x \times \eta_\omega$ avec $\eta_\omega < \eta_i * \eta_o$ pour (i) et de taille $M \times N = (\eta_i * \eta_o) \times \eta_\chi$ où $\eta_\chi < \eta_x$ pour (ii). Ces travaux, à travers les deux configurations présentées, montrent des résultats intéressants à analyser. En effet, pour un nombre de *clusters* ayant chacun le même nombre de vecteurs propres dans la base d'approximation, la factorisation des *ABRDFs* génère moins d'erreur que celle par texture. Cette observation montre que la cohérence des données est plus forte spatialement que dans le domaine angulaire et permet donc d'envisager des méthodes qui prennent en compte cet état de fait. L'avantage de la factorisation locale réside également dans les performances de la reconstruction des données lors du rendu. En effet, les matrices locales ont une taille relativement faibles et autorisent donc, pour la reconstruction, des algorithmes en temps réel. De plus, les ratios de compression sont très élevés avec la méthode de factorisation des *ABRDFs* et la taille d'une *BTF* compressée est environ de 20 Mo pour un faible résiduel. Enfin, notons que ces algorithmes de compression ont été parallélisés sur matériel graphique [RRK09] afin de compresser une *BTF* dans des temps raisonnables.

Ma *et al.* [MCT+05] adoptent une approche par niveau de détail pour décomposer une *BTF* par point de vue en utilisant l'ACP. Les données sont tout d'abord projetées sous forme de pyramide Laplacienne pour modéliser une *BTF* suivant plusieurs échelles de détails. Chaque *ABRDF* est ensuite décomposée par une ACP où seuls les K vecteurs propres les plus influents sont préservés. L'avantage de cette méthode réside dans la suppression de l'*aliasing* pendant la phase de rendu. Cependant, la compression des données, bien que satisfaisant, n'est pas optimale car la projection sous forme de pyramide Laplacienne ne décorréle pas les données : les données compressées sont donc redondantes.

Factorisation par tenseurs

Linéariser une image sous forme de vecteur colonne permet de créer une matrice contenant plusieurs images. En utilisant une ACP sur ce type de matrice, seules les colonnes peuvent être factorisées dans une base d'approximation formées par un nombre restreint de vecteurs propres. Par cette approche, les données redondantes suivant les lignes de la matrice sont donc conservées par la factorisation. Pour élargir la factorisation à l'ensemble des dimensions, certaines méthodes [VT04, WWS+05b, RK09] utilisent une approche par tenseur, généralisation d'une matrice à plusieurs dimensions, de manière à factoriser l'ensemble des données d'une *BTF*. Le principe consiste

à réaliser une DVS en plusieurs dimensions et ainsi modéliser les six dimensions de la *BTF* sous forme de produit de tenseurs. Ce type d'approche est très efficace en terme de compression car l'ensemble des dimensions de la *BTF* est factorisé par un nombre limité de composantes. Néanmoins, un telle méthode présente de nombreux inconvénients. Premièrement, les calculs pour effectuer la décomposition en tenseurs sont très longs, en dizaine d'heures sur une *BTF*. Ensuite, la taille mémoire nécessaire pour factoriser l'ensemble des dimensions d'une *BTF* en une décomposition multi-dimensionnelle est de l'ordre de plusieurs giga-octets : la *BTF* dans son entier et la mémoire temporaire nécessaire aux calculs. Une solution proposée par Wang *et al.* [WWS+05b] permet d'appliquer la méthode hors mémoire (*out-of-core*). Enfin, la reconstruction des données nécessite l'évaluation de multiple décomposition matricielle et ne permet pas un rendu temps réel d'une *BTF*.

Factorisation par NMF

Bien que non destinée à l'origine pour les *BTFs*, la *Non-negative Matrix Factorisation* (NMF) est une alternative intéressante aux méthodes précédemment expliquées. Le principe consiste à considérer une matrice V telle que $V \approx W.H$ où tous les coefficients de V, W et H sont positifs. Cette propriété n'est pas respectée dans les approches de type ACP mais révèle des caractéristiques indispensables pour l'édition de matériau. En effet, si la matrice V est décomposée suivant des règles adéquates, H devient une base de *BRDFs* (ou *ABRDFs*) visuellement compréhensible et donc éditable. Cette approche est reprise dans des travaux sur les *SVBRDFs* [LRR04, LBAD+06, PL07, WWHL07] où des matériaux issus de l'acquisition sont factorisés sous cette forme. D'après les auteurs [LBAD+06], le procédé est appliquable sur les *BTFs* également mais ce type de méthode nécessite des adaptations sur les règles de factorisation pour être utilisable avec les *BTFs*.

La NMF utilise des méthodes de programmation quadratique pour résoudre un système linéaire sous contrainte au sens des moindres carrés. Les textures directionnelles issues de l'acquisition sont linéarisées sous forme de matrice, V de taille $\eta_x \times (\eta_o * \eta_i)$. En choisissant au préalable le nombre de colonnes (resp. lignes) K de la matrice W (resp. H), le problème est défini comme un problème convexe en itérant les mises à jour successives de la matrice W puis de la matrice H individuellement. A chaque mise à jour de W ou de H, des contraintes linéaires d'inégalités ou d'égalités spécifiques à chaque matrices sont mises en place pour la minimisation des systèmes. En gardant des contraintes identiques aux modèles de *BRDF* pour la matrice H, réciprocité et conservation de l'énergie, cette matrice devient une base d'approximation de *BRDFs* qui peut être elle-même encore décomposée pour l'édition.

Ces méthodes permettent d'une part une compression des données d'une *SVBRDF*, voire d'une *BTF*, et d'autre part, l'édition de matériaux via les structures de données génèrées. Cependant, cette représentation n'est pas très efficace pour du rendu temps réel et souffre des mêmes limitations que les approches ACP pseudo-globales pour la complexité des opérations de reconstruction. De plus, cette factorisation décompose uniquement l'espace angulaire alors que la corrélation, bien que dépendante de l'échantillonnage, est souvent plus forte dans les dimensions spatiales. Enfin, le processus permet de converger uniquement vers un minimum local ce qui implique une initialisation du système rigoureuse.

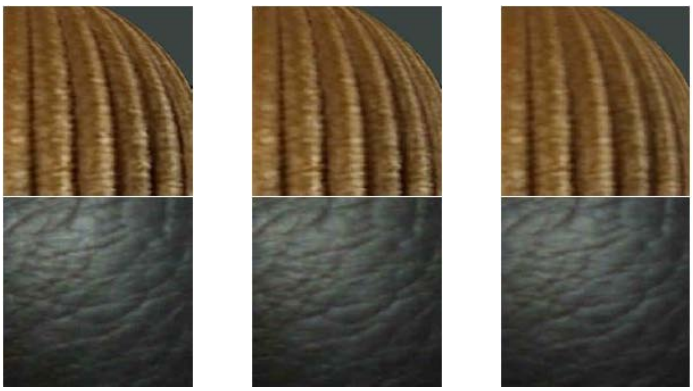
2.2.2.3 Discussion

La décomposition d'une *BTF* dans une base linéaire se révèle particulièrement efficace pour la compression. En effet, la taille des données ainsi factorisées est de l'ordre de 20 Mo pour les méthodes les plus efficaces avec une erreur presque imperceptible lors du rendu induite par un lissage des dimensions factorisées. Ces méthodes de compression ont de loin le meilleur compromis entre la taille des données compressées et l'erreur d'approximation.

La cohérence dans l'espace des directions est bien exploitée mais celle dans les dimensions spatiale est trop souvent mise de coté. En effet, seule l'approche ACP par *clusters* [MMK03b] autorise une factorisation de l'espace texture et cette approche génère les meilleurs résultats de compression à ce jour. Notons que les méthodes basées tenseurs permettent de décorréler l'ensemble des dimensions d'une *BTF* mais posent encore trop de problèmes numériques et une taille en mémoire trop importante pour être utilisables. Il est donc nécessaire de conjuguer avec ce type d'approche une décorrélation spatiale pour d'une part, améliorer encore les taux de compression et d'autre part, fournir une approche multi-résolution pour accélérer les calculs et agrémenter la *BTF* par des niveaux de détail comme dans [MCT+05]. D'autres techniques de factorisation sont aussi possibles pour améliorer les résultats comme celle présentée par Wu *et al.* [WDR09] qui utilisent des factorisation suivant les lignes et suivant les colonnes d'une matrice [PWS06].

Enfin, le rendu en temps réel des données compressées est laborieux. En effet, la complexité des calculs de reconstruction est souvent trop importante pour fournir des algorithmes en temps réel. Cette observation se dissimule suivant deux principes lors de la reconstruction : le nombre d'accès texture, encore coûteux sur le matériel graphique actuel, et le nombre d'opérations dû aux multiplications vectorielles pour la reconstruction d'un pixel. Ici encore, une approche multi-résolution permet de diminuer la masse des calculs comme exposé dans [MCT+05].

2.2.3 Compression par substitution



(a) (b) (c)

FIGURE 2.10: Sous-échantillonnage de *BTFs* [FCH08]. (a) original avec $\eta_o = 81$ et $\eta_i = 81$. (b) $\eta_o = 41$ et $\eta_i = 41$. (c) $\eta_o = 31$ et $\eta_i = 31$.

Récemment, Filip *et al.* [FCH08, FCGH08, FCH09] ont proposé des méthodes de compression basées perception qui présentent un processus de factorisation par substitution. En effet, comme évoqué dans la partie 2.1, les systèmes d'acquisition ne peuvent pas échantillonner de manière adaptative l'espace des directions suivant le type de matériau à capturer. De ce fait, l'échantillonnage est identique quelque-soit le matériau mais suffisamment dense pour capturer la plupart des effets directionnels d'une large classe de matériaux. Dans un premier temps, les travaux de Filip *et al.* [FCH08, FCH09] ont identifié l'impact perceptuel généré par un sous-échantillonnage du domaine angulaire des *BTFs* (voir Figure 2.10). Cette étude a montré que l'espace des directions du point de vue était plus propice, lors d'un ré-échantillonage, à une gêne perceptuelle que l'espace des directions de la lumière. De même, le ré-échantillonage suivant l'angle d'élévation est plus critique que celui suivant l'angle azimutal. Dans [FCGH08], les auteurs ont montré que, plutôt que de sous-échantillonner les données, une hiérarchie perceptuelle peut être mise en place dans l'espace des directions où un échantillon qui n'a pas ou peu de différences perceptuelles avec son voisin peut être substitué par ce dernier. Cette méthode permet de limiter le nombre de textures directionnelles à compresser et, en utilisant la méthode de Müller *et al.* [MMK03b] sur une *BTF* en partie substituée, a montré un gain de compression par quatre. Néanmoins, bien que validé par une étude utilisateur, le rendu de ces travaux révèlent des artefacts visuels gênants pour un utilisateur averti : effets de parallaxes parfois brutaux lors d'un changement de point de vue ou interpolation des ombres approximatives.

2.2.4 Bilan

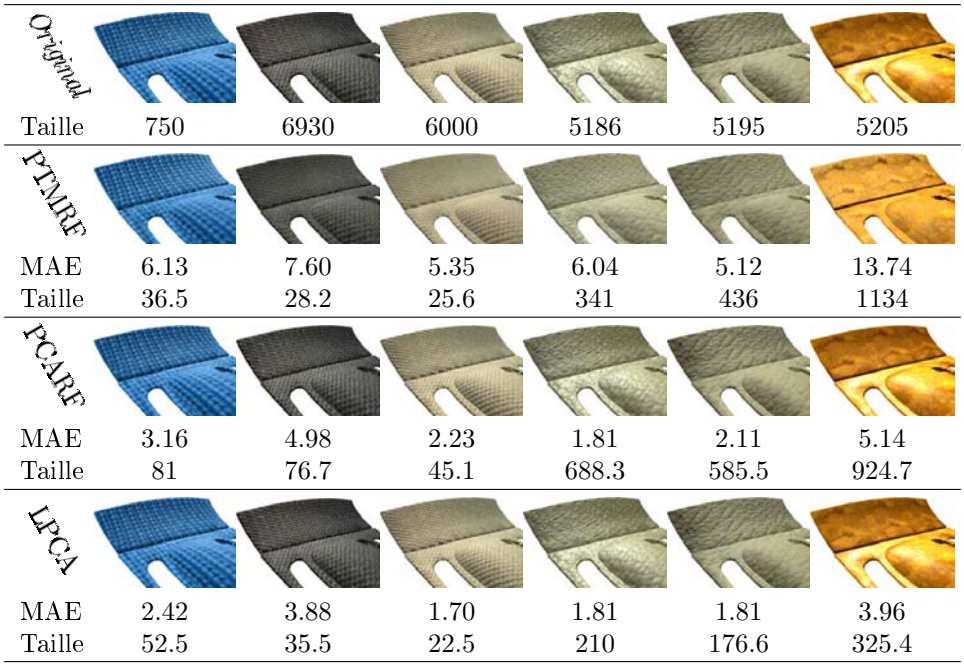


FIGURE 2.11: Résultats de quelques méthodes de compression de *BTFs* issues de [FH08]. L'erreur moyenne MAE est calculée dans l'espace CIE Lab et la taille de données est en Mega-octet. La méthode PTMRF correspond aux travaux de [MMK03a], PCARF à [SSK03] et LPCA à [MMK03b].

Les résultats visuels de méthodes de l'état de l'art, extraits de [FH08], sont présentés sur la Figure 2.11. Bien que non exhaustives, ces indications exposent le comportement général des travaux présentés dans cette section. Les méthodes basées approximation souffrent d'erreurs d'approximation dues aux nombreux effets inclus dans une *BTF* et proposent un taux de compression perfectible. Cette observation est visible sur la Figure 2.11 avec la méthode PTMRF où les reflets spéculaires sur certains matériaux sont lissés. Cependant, l'approximation par un modèle de *BRDF* ou un polynôme bivarié par pixel permet un rendu temps réel d'une *BTF*. A l'inverse, les méthodes basées factorisation ont un taux de compression très élevé tout en restant fidèles aux données originales. Les méthodes PCARF et LPCA sur la Figure 2.11 soulignent ce constat : il est difficile de les différencier de l'original. En revanche, l'utilisation de telles méthodes pour une application en temps réel reste problématique. Enfin, notons que l'approche par substitution n'est pas une méthode de compression en soi mais permet d'effectuer un échantillon-

nage psycho-visuel adaptatif en pré-traitement. Un tableau récapitulatif de l'efficacité de chaque méthode est présenté sur la Table 2.1.

	Temps de calcul	Compression/Qualité	Efficacité Rendu
Approximation			
BRDF	⋆	⋆	⋆⋆⋆
Polynomial	⋆⋆⋆	⋆	⋆⋆⋆
Factorisation			
ABRDF	⋆⋆	⋆⋆	⋆⋆
ACP	⋆	⋆⋆⋆	⋆(⋆) [a]
Tenseur	⋆	⋆⋆⋆	⋆
NMF	⋆	⋆⋆	⋆
Substitution			
Fixe	⋆⋆	⋆⋆	*n/a*
Adaptatif	⋆⋆	⋆	*n/a*

a. approche par cluster

TABLE 2.1: Efficacité des méthodes de compression d'après la classification proposée. « ⋆ » correspond au niveau le moins bon alors que « ⋆⋆⋆ » est le niveau d'efficacité maximal. Le temps de calcul est le temps nécessaire pour compresser les données. La colonne « Compression / Qualité » juge du taux de compression par rapport à l'erreur introduite. La dernière colonne expose la complexité des calculs de reconstruction des données pour un rendu en temps réel.

D'un point de vue général, un atout important en informatique graphique est manquant dans l'ensemble des méthodes de compression existantes : la multi-résolution. En effet, mis à part [MCT+05] qui propose une méthode avec niveaux de détails spatiaux, aucun des travaux existants étudie l'aspect multi-échelle d'une *BTF*. Or, fournir des échelles de représentation pour une *BTF* est préférable pour plusieurs raisons. Premièrement, la diminution des temps de calcul lors de la compression. En effet, de nombreux travaux annexes adoptent une analyse multi-résolution pour résoudre des problèmes complexes afin de faciliter les traitements. Un exemple concret ici serait, par exemple, l'initialisation des paramètres d'un modèle paramétrique de *BRDF* pour une approche par approximation où la recherche est simplifiée si l'échelle choisie est suffisamment grossière. Le processus de minimisation pourrait être guidé par échelle, de l'échelle grossière vers le niveau le plus fin, pour converger vers un minimum proche du minimum global. Ensuite, la compression des données est, la plupart du temps, réalisée soit suivant l'espace angulaire soit suivant les dimensions spatiales. Une représentation multi-échelle dans l'espace non compressé permettrait de décorréler les don-

nées, à condition de choisir la bonne transformation, et ainsi améliorer encore les taux de compression. Enfin, lors du rendu, il est indispensable de fournir des échelles de détail pour éviter les phénomènes de recouvrement. Certes, le matériel graphique actuel permet le filtrage de texture mais cette approche a un coût non négligeable. Une méthode de compression multi-échelle permettrait donc de reconstruire uniquement les données nécessaires lors du rendu.

2.3 Edition et Synthèse



(a) (b)

FIGURE 2.12: Edition de *BTFs*. (a) [MSK07] matériau original (en haut) avec sa méso-structure modifiée (en bas). (b) [KBD07] matériau original (à gauche) avec édition de la couleur (à droite).

L'édition de *BTFs* est un domaine d'étude récent pour lequel peu de travaux ont été réalisés. Deux approches sont présentes dans l'état de l'art. Les travaux de Müller *et al.* [MSK07] portent sur l'extraction de comportements clés de la méso-structure du matériau dans l'espace des directions pour synthétiser une nouvelle méso-géométrie, dont les caractéristiques directionnelles seront identiques au matériau d'origine. Cette approche permet, par exemple, d'effectuer des déformations au niveau de la méso-structure comme illustré sur la Figure 2.12 (a). Ces travaux autorisent, pour un même matériau, une pluralité d'aspects : un lainage avec des mailles différentes, des rainures de cuir avec des trajectoires diverses, *etc.* A noter que ce type d'édition nécessite le calcul peu robuste d'éléments géométriques de la méso-structure (normales, hauteur) qui sont extraits ici à l'aide de méthodes stéréoscopiques. Enfin, la méthode présentée ici est réalisée hors ligne, vue la complexité des calculs engendrés.

Dans [KBD07], Kautz *et al.* présentent une application complète d'éditions de *BTFs* : *BTFShop*. Ces travaux, inspirés des logiciels de manipulation d'images, regroupent des boîtes à outils permettant de modifier un très grand nombre de paramètres d'une *BTF* : couleur, hauteur, ombrage, parallaxe, *etc.* L'édition devient donc ici un travail interactif dans lequel l'utilisateur peut modifier, la plupart du temps en temps réel, les propriétés d'une *BTF*. Cette faculté d'interaction est possible grâce à un découpage de la *BTF* en blocs de données modifiables indépendamment et ainsi, éviter de charger les giga-octets d'une *BTF* dans son ensemble en mémoire. Les nombreux outils inclus dans *BTFShop* sont présentés en détail dans [KBD07] et un exemple est montré sur la Figure 2.12 (b). Enfin, l'application est disponible publiquement et utilisable gratuitement [BTFb].

La synthèse de *BTFs*, inspirée par les méthodes de synthèses d'images, consiste à générer un matériau, statistiquement équivalent au matériau d'origine, qui peut être répété à l'infini sur une surface sans répétition de motifs discernables. En découpant la *BTF* en morceaux élémentaires, ces méthodes peuvent être utilisées pour la compression [WD04, FH05, FH07, Hai07]. Une étude complète de ces approches est référencée dans les revues de l'état de l'art sur les *BTFs* [MMS+04, FH08].

Chapitre 3

Les ondelettes

Les ondelettes se présentent comme une alternative efficace à l'analyse de Fourier. Cette dernière permet de définir le *spectre* d'un signal mais ne donne que très peu d'information sur la localisation des différentes composantes fréquentielles. Les analyses multi-résolutions permettent de déterminer des fonctions de bases qui sont à la fois localisées en temps et en fréquence sur lesquelles est projeté un signal. L'idée de la décomposition en ondelettes est de projeter une fonction ou un signal sur des fonctions de bases obtenues par translation et dilatation de fonctions de référence telle que celle tracée sur la Figure 3.1.

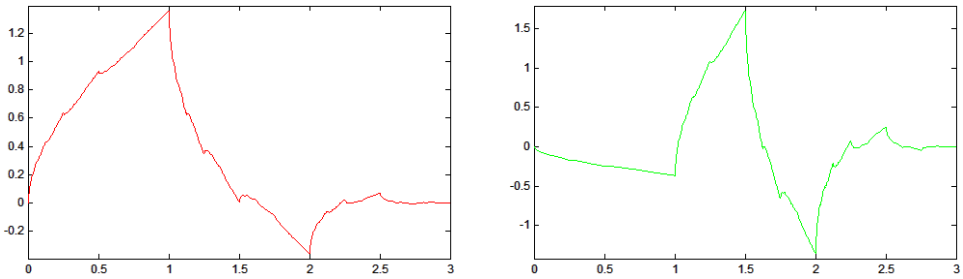


Figure 3.1: Exemple de fonction d'échelle (à gauche) et d'ondelettes (à droite), les *D4*, proposées par Ingid Daubechies [Dau92].

3.1 Ondelettes et analyse multi-résolution

Avec la théorie de la multi-résolution [Mal89], l'ondelette ne constitue plus l'élément majeur de la décomposition. Les ondelettes jouent toutefois un rôle essentiel dans la mesure où elles encodent la différence entre deux niveaux de résolution successifs. Si la distance entre deux échantillons est constante, l'analyse multi-résolution (AMR) est réalisée par translation et

dilatation d'une fonction d'échelle telle que :

$$\varphi_i^j(x) := \varphi(2^j x - i),\ i = 0, \ldots, 2^j - 1.$$

Une fonction d'échelle, φ_i^j, fournit une représentation du signal à une résolution qui diffère de la précédente par un facteur deux. L'espace V^0 est l'espace engendré par la fonction d'échelle φ_i^0. L'espace V^j est engendré par les fonctions de V^0 comprimées d'un facteur 2^j. A une résolution donnée le signal doit contenir toute l'information du signal aux résolutions plus grossières. On peut ainsi noté que :

$$V^0 \subset V^1 \subset V^2 \subset \ldots$$

De telles caractéristiques amènent à définir des propriétés fortes sur les fonctions $\varphi(x)$ dont les principales, dans le cas général, sont :

- support compact,
$$\varphi_i^j(x) = 0 \text{ si } x_p = p * 2^{-j}, p \neq i.$$
- interpolation,
$$\varphi_i^j(x) = 1 \text{ si } x_p = p * 2^{-j}, p = i.$$
- raffinement,
$$\varphi_i^j(x) = \sum_l h_l * \varphi_l^{j+1}(x).$$
où les h_l sont appelés *filtres*.

L'espace W^j associé à l'ondelette est, généralement, orthogonal à l'espace V^j et représente la différence entre V^j et V^{j+1} :

$$W_i \oplus V_i = V_{i+1}$$

Les fonctions linéairement indépendantes, ψ_i^j, définissant l'espace W^j sont appelées *ondelettes*. Les ondelettes sont donc définies telles que :

$$\psi_i^j(x) := \psi(2^j x - i),\ i = 0, \ldots, 2^j - 1.$$

et sont le complément à V^j dans V^{j+1}. Puisque $W^j \subset V^{j+1}$, il existe un ensemble de g_l tels que,

$$\psi_i^j(x) = \sum_l g_l * \varphi_l^{j+1}(x)$$

et qui décrivent les fluctuations.

Si on définit W^j comme le complémentaire *orthogonal* à V^j dans V^{j+1}, les W^j sont orthogonaux entre eux et on parle alors d'AMR *orthogonale*. Mathématiquement, la définition d'une base d'ondelettes orthogonale multirésolution est formalisée suivant :

$$\left.\begin{array}{rcl} < \varphi_k^j, \varphi_l^j > & = & \delta_{k,l} \\ < \psi_k^j, \psi_l^j > & = & \delta_{k,l} \\ < \varphi_k^j, \psi_l^j > & = & 0 \end{array}\right\} \forall j, k, l.$$

Cependant, l'orthogonalité stricte de la base d'ondelettes est restrictive quant à la forme des ondelettes. Dans certains cas, la symétrie centrale est une propriété importante pour les fonctions d'échelle et les ondelettes. Il n'existe pas de bases d'ondelettes, mise à part la base de Haar, qui est à la fois orthogonale, symétrique et à support compact. En relâchant la contrainte d'orthogonalité, une base d'ondelettes dite *semi-orthogonale* peut être construite où seule la condition suivante est respectée :

$$< \varphi_k^j, \psi_l^j >= 0, \forall j, k, l.$$

c'est à dire, que les fonctions d'échelle sont orthogonales aux ondelettes mais que les fonctions d'échelle (resp. les ondelettes) ne sont plus orthogonales les unes aux autres. Une telle propriété facilitent la construction de bases d'ondelettes à l'aide de fonctions symétriques telles que les *B-Splines*.

Plus récemment, avec l'apparition des ondelettes *biorthogonales* [CDF92] et du *lifting scheme* [Swe96], une grande variété de bases d'ondelettes peut être construite en ne considérant plus l'orthogonalité comme une contrainte entre les fonctions d'échelle et les ondelettes mais via une base duale. Dans ce cas, il existe des fonctions d'échelle duales $\tilde{\varphi}_i^j$ et ondelettes duales $\tilde{\psi}_i^j$ qui génèrent une AMR duale telle que :

$$\left.\begin{array}{rcl} < \varphi_k^j, \tilde{\psi}_l^j > & = & 0 \\ < \psi_k^j, \tilde{\varphi}_l^j > & = & 0 \end{array}\right\} \forall j, k, l.$$

De plus, une base d'ondelettes *biorthogonale* doit satisfaire également les propriétés suivantes :

$$\left.\begin{array}{rcl} < \varphi_k^j, \tilde{\varphi}_l^j > & = & \delta_{k,l} \\ < \psi_k^j, \tilde{\psi}_l^j > & = & \delta_{k,l} \end{array}\right\} \forall j, k, l.$$

Cette évolution sur la flexibilité de la forme des bases d'ondelettes conjointement à une construction facilitée par la méthode du *lifting scheme* [Swe97] a permis l'émergence d'une nouvelle génération d'ondelettes, ondelettes de deuxième génération ou ondelettes géométriques, dont le cadre applicatif n'est plus restreint à un domaine régulier. En effet, ces méthodes ont permis de définir une décomposition en ondelettes sur des maillages [LDW97, EDD+95, CPD+96] ou plus particulièrement sur la sphère [SS95, LF08].

3.2 Cas pratiques

3.2.1 Ondelettes de première génération

Considérons a et b, deux réels consécutifs dans un signal 1D où les échantillons sont distribués de manière régulière. Le but des ondelettes est d'exploiter la corrélation entre les valeurs d'un voisinage pour exprimer un signal

avec le moins de coefficients possibles. Un moyen simple d'utiliser cette propriété est de représenter a et b par leur moyenne s et leur différence d :

$$\begin{aligned} s &= \frac{a+b}{2} \\ d &= b-a \end{aligned}$$

Ainsi, si a et b sont fortement corrélés alors la valeur absolue de la différence sera proche de 0. La transformation est sans perte dans la mesure où les coefficients peuvent être retrouvés de la manière suivante :

$$\begin{aligned} a &= s - d/2 \\ b &= s + d/2 \end{aligned}$$

Cette approche correspond à la base d'ondelettes la plus simple : la base de Haar. Considérons maintenant l'analyse du signal conjointement à l'AMR avec un signal s_n de 2^n échantillons, $s_{n,i}, 0 \leq i < 2^n$. La méthode précédente peut être appliquée sur chaque couple avec $a = s_{n,2i}$ et $b = s_{n,2i+1}$ où $0 \leq i < 2^{n-1}$. On obtient ainsi, pour tout i,

$$\begin{aligned} s_{n-1,i} &= \frac{s_{n,2i} + s_{n,2i+1}}{2} \\ d_{n-1,i} &= s_{n,2i+1} - s_{n,2i} \end{aligned}$$

Le signal s_n contenant 2^n échantillons est donc séparé en deux signaux de taille 2^{n-1}, s_{n-1} approximation du signal d'origine et d_{n-1} encodant la différence entre s_n et s_{n-1}. Le procédé est répété récursivement sur le signal d'approximation jusqu'à s_0, contenant un seul élément, la moyenne de tous les échantillons de s_n. Le signal s_n peut donc être reconstruit à différentes échelles à partir de s_0 et des coefficients de détail $d_0, \ldots, d_{n-1}$.

Utiliser la décomposition en ondelettes à l'aide d'une base de Haar peut donc se traduire par l'utilisation de filtres convolutifs, méthode appelée *filter bank*, où les filtres sont définis tels que :

	Analyse	Synthèse
approximation	[0,5 ;0,5]	[1 ;-0,5]
détail	[-1 ;1]	[1 ;0,5]

De nombreuses bases d'ondelettes ont été développées en utilisant cette approche. Dans ce manuscrit, seule la famille des bases de Daubechies [Dau92] est utilisée, ce qui inclue la base de Haar et un ensemble d'ondelettes de degrés différents.

L'utilisation des ondelettes sur une image est devenue un traitement classique en analyse d'images. Les transformations présentées ici sont la généralisation en deux dimensions de la transformation unidimensionnelle par simple produit tensoriel. Une décomposition 2D dite standard consiste à appliquer

dans un premier temps une transformation 1D pour tous les niveaux sur les lignes de l'image puis sur les colonnes. La décomposition non-standard consiste à alterner la décomposition sur les lignes et sur les colonnes à chaque niveau de décomposition. Cette dernière sera privilégiée dans les travaux présentés dans ce manuscrit.

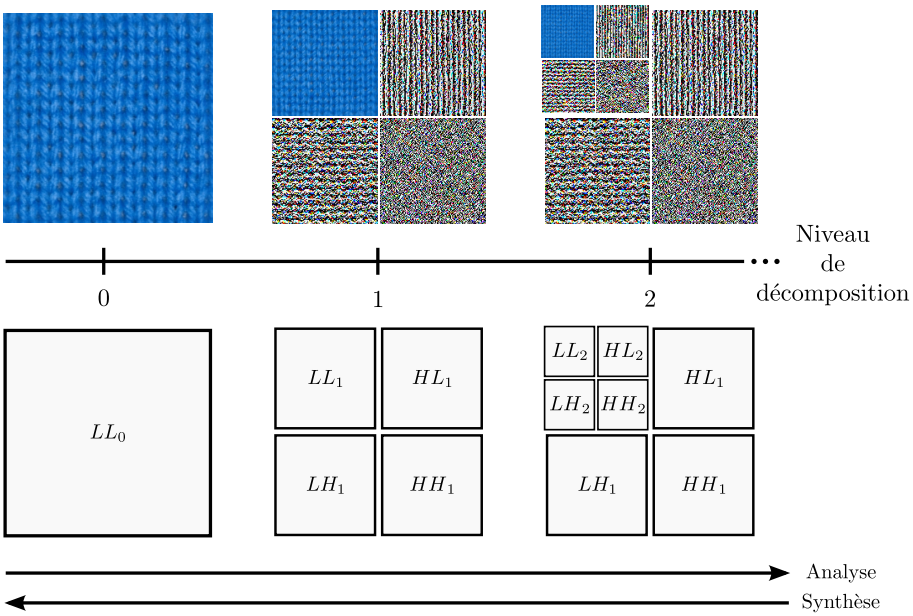


FIGURE 3.2: Décomposition non-standard d'une image dans une base d'ondelettes de Haar. L'image est subdivisée en sous-bandes : une sous-bande d'approximation LL_j représentant l'approximation du signal d'origine au niveau j ; trois sous-bandes de détail HL_j, LH_j et HH_j représentant les détails entre deux niveaux d'approximation dans le sens, respectivement, horizontal, vertical et diagonal.

La Figure 3.2 illustre une décomposition en ondelettes d'une image à l'aide d'une décomposition non-standard. La premier niveau de décomposition sépare les coefficients en deux classes : les coefficients d'approximation (LL_j) et les coefficients de détail (HL_j, LH_j et HH_j). Seuls les coefficients d'approximation sont par la suite décomposés récursivement jusqu'à un niveau donné.

3.2.2 Ondelettes géométriques

Les ondelettes géométriques appliquées aux maillages triangulaires sont intimement liées aux surfaces de subdivision. Comme illustré sur la Figure 3.3, un maillage de base est subdivisé jusqu'au niveau de détail souhaité : cette approche correspond à la notion de fonctions d'échelle qui dépendent

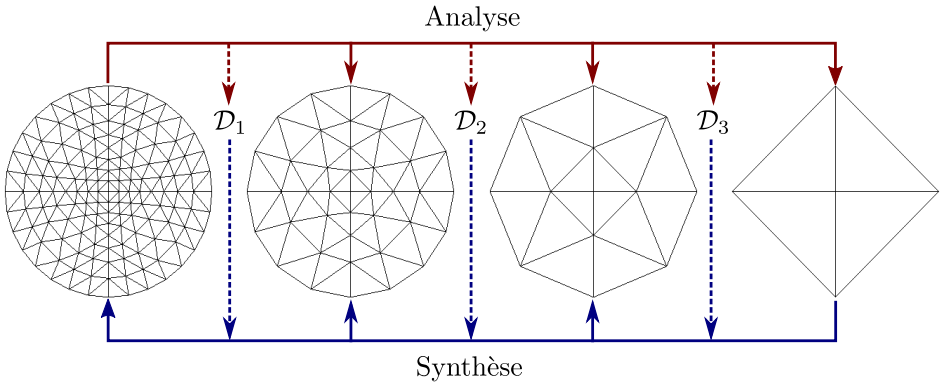


FIGURE 3.3: Décomposition d'un hémisphère triangulé dans une base d'ondelettes (vue projective). A chaque niveau de décomposition j, le nombre de faces de l'hémisphère varie d'un facteur 4. Les détails D_j représentent la différence entre deux niveaux de décomposition et peuvent correspondre à des détails géométriques ou des propriétés définies en chaque point du maillage.

du schéma de subdivision employé (ici, un simple schéma un vers quatre). A partir de ces fonctions d'échelle, les ondelettes sont définies telles que les détails entre deux niveaux de subdivision, les D_j, soient capturés par ces fonctions.

Généralisé à des surfaces de topologie arbitraire [LDW97, EDD+95], le maillage de base est calculé par des techniques de maillage progressif [Hop96, GZ05]. Le maillage de base est par la suite raffiné par subdivision adaptative où les coefficients de détails de l'analyse en ondelettes correspondent à la perturbation géométrique des points générés par un schéma de subdivision pour correspondre au maillage originel. Notons que tout attribut défini sur un maillage (couleur en chaque point, orientation des normales, ...) peut être ainsi décomposé.

D'autres méthodes utilisent les ondelettes sur la sphère [SS95, LF08] dont leur principal intérêt est d'encapsuler un comportement directionnel. Les ondelettes sphériques ont été utilisées comme base pour la projection de *BRDFs* ou de cartes d'environnement [OS95]. La Figure 3.4 décrit l'analyse en ondelettes sur un cas simple. Le maillage de base est un triangle auquel un niveau de subdivision est appliqué (schéma de subdivision un vers 4). Chaque point, ou attribut associé au point, est projeté dans la base d'ondelettes d'une manière différente suivant si le point est un coefficient d'approximation au niveau 1 ou un coefficient de détail. Les coefficients de détail, issus de la subdivision, sont projetés dans une base linéaire d'ondelettes par soustraction pondérée de ses points voisins. Les coefficients d'approximation sont invariants lors de la projection dans cette base simple. Ce cas correspond à une décomposition linéaire par point proposé par Peter Schröder et Wim

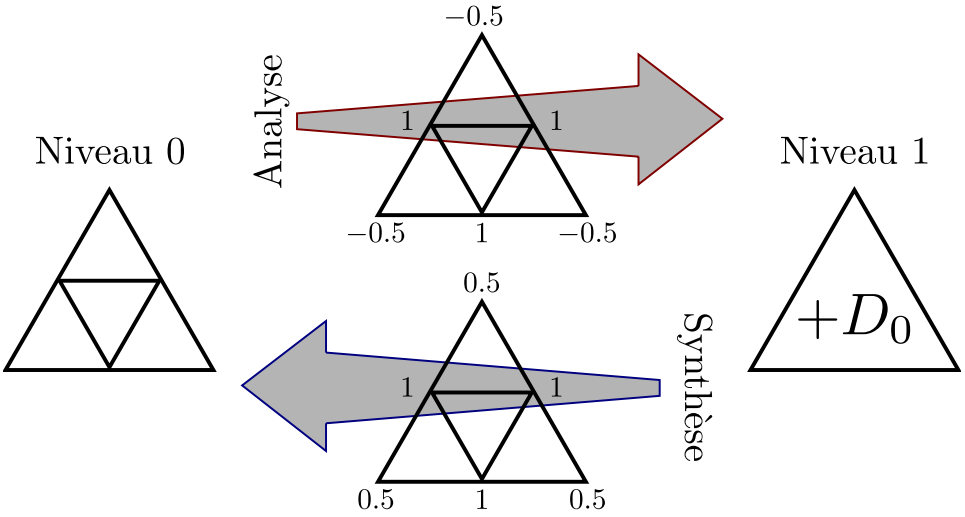


FIGURE 3.4: Schéma d'analyse et de synthèse pour la projection de points d'un maillage dans une base d'ondelettes simple [SS95]. Les coefficients appliqués à chaque point du maillage lors de l'analyse (resp. la synthèse) sont spécifiés pour générer (resp. reconstruire) les coefficients de détails, D_0, entre deux niveaux de décomposition. La pondération est effectuée par arête du maillage et les coefficients d'approximation sont inchangés d'un niveau à l'autre.

Sweldens dans [SS95] appliquée, normalement, sur un maillage sphérique. Bien que triviale, cette approche reste efficace et permet une faible complexité lors de la reconstruction, idéale pour des algorithmes de rendu.

3.3 Compression par ondelettes

3.3.1 Intuition

Une décomposition en ondelettes ne définit pas en soit un algorithme de compression mais une étape judicieuse pour ce type d'application. En effet, l'analyse en ondelettes possède de telles propriétés de décorrélation que de nombreux coefficients de détail ont une très faible amplitude. On se retrouve alors avec une représentation *creuse* : la décomposition en ondelettes se révèle donc comme un candidat idéal pour la compression. En supposant que l'on dispose d'une représentation compacte encodant uniquement les coefficients non nuls, les taux de compression théoriques apparaissent comme très performants dans une approche sans perte.

Si l'on observe la Figure 3.5, on constate que la différence dans l'histogramme des coefficients avant et après transformation en ondelettes est

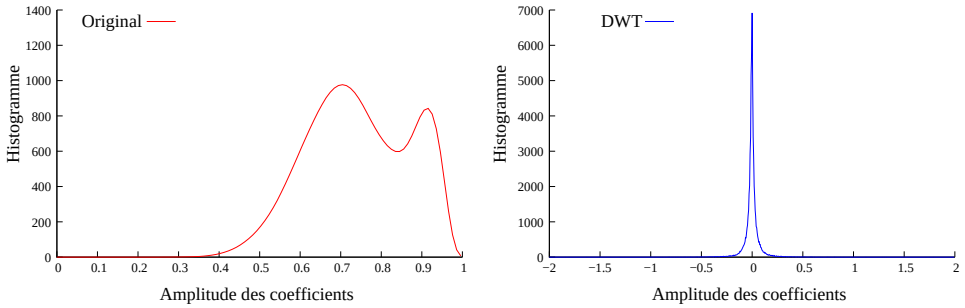


FIGURE 3.5: Histogramme des coefficients sur la luminance d'une image de taille 256x256. A gauche, l'histogramme des coefficients originaux normalisés entre 0 et 1. A droite, l'histogramme des coefficients d'ondelettes dans une base de D4 (zoom sur la partie significative).

très différente. Dans le cas de l'image originale, la valeur de la luminance des pixels est répartie sur l'ensemble du spectre. En revanche, après décomposition en ondelettes, la majorité des coefficients est située autour de la valeur 0, plus de 90% dans le domaine $[-0,2; 0,2]$. En effectuant un seuillage doux sur les coefficients, un très grand nombre des coefficients sera considéré comme non représentatif pour le signal et donc exclu du signal compressé.

Considérons une fonction f, d'énergie finie, définie comme une projection dans une base d'ondelettes orthogonale notée $\{u_1(x), \ldots, u_m(x)\}$:

$$f(x) = \sum_{i=1}^{m} c_i.u_i(x).$$

Le but ici est de représenter la fonction f avec moins de m coefficients tout en maîtrisant l'erreur d'approximation introduite dans la nouvelle fonction $\bar{f}$. Ce processus peut être réalisé en définissant une permutation π telle que,

$$c_{\pi(0)} \geq c_{\pi(1)} \geq \ldots \geq c_{\pi(m)},$$

et de choisir les $\bar{m}$, $\bar{m} < m$, coefficients de plus fortes amplitudes pour représenter une approximation de la fonction f :

$$f(x) = \sum_{i=1}^{\bar{m}} c_{\pi(i)}.u_{\pi(i)}(x).$$

Ainsi, le carré de l'erreur L^2 pour cette approximation est donné par :

$$\begin{aligned}
\|f(x) - \bar{f}(x)\|_2^2 &= < f(x) - \bar{f}(x), f(x) - \bar{f}(x) > \\
&= \left\langle \sum_{i=\bar{m}+1}^{m} c_{\pi(i)}.u_{\pi(i)}, \sum_{j=\bar{m}+1}^{m} c_{\pi(j)}.u_{\pi(j)} \right\rangle \\
&= \sum_{i=\bar{m}+1}^{m} \sum_{j=\bar{m}+1}^{m} c_{\pi(i)} c_{\pi(j)} < u_{\pi(i)}, u_{\pi(j)} > \\
&= \sum_{i=\bar{m}+1}^{m} \left(c_{\pi(i)}\right)^2
\end{aligned}$$

car la base est orthogonale : $< u_i, u_j >= \delta_{i,j}$. Ce résultat indique que le carré de l'erreur L^2 de l'approximation est la somme quadratique des coefficients qui sont exclus de la représentation. Donc, le meilleur choix pour minimiser l'erreur d'approximation est donc d'exclure les faibles coefficients en premier, ce qui justifie l'approche du seuillage des coefficients. Toute la difficulté d'une compression en ondelettes se résume donc au nombre de coefficients à exclure et au stockage de cette représentation creuse.

3.3.2 Codage *Zero-Tree*

La méthode dite du codage *Zero-Tree* (ZTC) est le moyen le plus efficace actuellement pour représenter une décomposition en ondelettes de la manière la plus compacte possible. L'idée fut introduite par Shapiro dans [Sha93] dans le contexte de compression d'images par ondelettes et se base sur deux hypothèses fondamentales quant à l'arborescence d'une décomposition en ondelettes :

- les images ont un spectre basse fréquence,
- les coefficients de plus fortes amplitudes sont les plus importants.

La première hypothèse consiste à supposer que l'énergie d'une sous-bande, la somme quadratique des coefficients d'une sous-bande, décroît à mesure que l'échelle augmente. En d'autres termes, les coefficients représentant les échelles basses fréquences sont, en valeur absolue, plus grands que ceux des échelles hautes fréquences. Le seconde hypothèse est directement liée aux explications présentées dans la partie 3.3.1 et laisse présumer une approche par plan de bits où les bits de poids forts seront prioritaires par rapport à ceux de poids faibles. Ces deux hypothèses sont exploitées dans le ZTC par un codage progressif des données ce qui permet d'une part, de décoder les données à une résolution privilégiée et d'autre part, de choisir un taux de compression en fonction de la longueur du code conservé.

L'idée centrale du ZTC est d'exploiter la dépendance inter-échelles des coefficients d'ondelettes pour effectuer le codage. Des arbres de coefficients sont ainsi définis et, dans le cas idéal guidé par les hypothèses énoncées

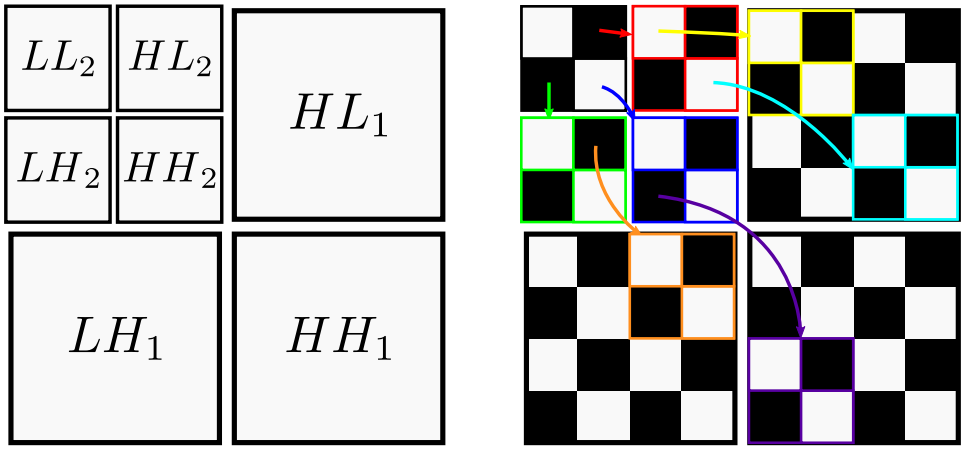


FIGURE 3.6: Arbre de coefficients liés à travers les échelles (non exhaustif). Un coefficient au niveau $j+1$ est lié à quatre coefficients au niveau j.

précédemment, seront codés en une seule fois. Concrètement, un coefficient de détail au niveau $j+1$ est lié à quatre coefficients, dans la même direction, au niveau j et ainsi de suite jusqu'au niveau 1. Cette arborescence est représentée sur la Figure 3.6 et définit la décomposition en ondelettes comme un arbre quaternaire. Cette parentalité est exploitée dans le ZTC par le fait que l'amplitude des coefficients décroît à mesure que l'échelle s'affine. Ainsi, lorsqu'un coefficient de la racine est inférieur à un certain seuil, les autres coefficients de la hiérarchie sont aussi inférieurs à ce seuil et l'arbre entier peut être codé par un unique symbole. Un *Zero-Tree* est donc un arbre quaternaire dont tous les coefficients sont inférieurs, en valeur absolue, au coefficient racine.

L'algorithme du ZTC effectue plusieurs approximations successives par quantification des coefficients d'ondelettes. Cette quantification est réalisée en codant la décomposition en ondelettes par plan de bits, du bit de poids fort vers le bit de poids faible, et affiner ainsi la décomposition à chaque passe. Sur chaque plan de bits, une première étape consiste à classer les coefficients pour être éventuellement codés s'ils sont des *Zero-Tree*. La seconde étape est une étape de raffinement où les coefficients non-codés durant la première passe seront codés d'une manière particulière.

D'autres méthodes ont été développées à partir de cette philosophie de codage pour les images [SP96, PAB00, Tau00] ou les maillages [LDW97]. Ces travaux diffèrent par la manière de parcourir les coefficients et sur la gestion des arbres qui ne sont pas de *Zero-Tree*. Notons que ce type de codage est employé dans la norme JPEG2000 [JPE00], méthode de compression d'images la plus aboutie à ce jour.

3.3.3 Codages typés matériel graphique

Bien que représentant un codage très performant pour une décomposition en ondelettes, le codage *Zero-Tree* souffre d'un inconvénient majeur pour tous les algorithmes d'informatique graphique : l'adaptation au matériel graphique. En effet, le ZTC est représenté par un flux de bits ; pour reconstruire une image ou un maillage, ce flux de bits est parcouru du début jusqu'au niveau de détail souhaité en analysant chaque bit et nécessite des branchements conditionnels fréquents. Dans un contexte massivement parallèle comme la programmation sur matériel graphique, ce concept peut être implémenté par l'utilisation de plusieurs passes de rendu où une passe correspond au raffinement de la passe précédente. Cependant, un algorithme multi-passe sur matériel graphique est coûteux et cette approche serait donc inefficace. De nouvelles méthodes de programmation graphique [Cud] pourraient être envisagées par l'utilisation de synchronisations et de mémoires partagées entre les différents processus en parallèle. Néanmoins, l'intérêt d'une telle approche est discutable car les performances exigées pour un rendu en temps réel, à mon avis, ne pourraient pas être atteintes.

Dans ce contexte, les algorithmes utilisant une représentation compressée d'une décomposition en ondelettes sont très rares en informatique graphique. Cela ne veut pas dire que les ondelettes ne sont pas utilisées, par exemple [KTHS06, WLHW07, TP08], mais rarement sous forme compacte. Récemment, Mike Boulton [Bou08] a exposé quelques méthodes pour utiliser des signaux à base d'ondelettes, sous forme compressés, sur matériel graphique. Cette présentation a mis en avant la possibilité d'utiliser une représentation compressée à base d'ondelettes directement sur les processeurs graphiques programmables mais a surtout montré la pauvreté des algorithmes en la matière. Cependant, quelques méthodes utilisant ce concept sont présentes dans la littérature, souvent ad-hoc, mais suffisamment innovantes pour en décrire le fonctionnement.

Dans un contexte d'éclairage globale, Hašan *et al.* [HPB06] proposent de factoriser un ensemble de textures, où chaque texture représente la contribution de chaque lumière dans une scène 3D, par une texture d'*atlas*. Dans leur méthode, les coefficients d'ondelettes non-nuls sont situés de manière cohérente et localisés dans des régions spatiales facilement identifiables. Les régions à coefficients non-nuls sont représentées par un rectangle englobant. Si le nombre de coefficients nuls à l'intérieur de ce rectangle est supérieur à un seuil fixé alors le rectangle est subdivisé en plusieurs sous-rectangles jusqu'à ce que le critère de densité soit atteint. Cette segmentation des données délivre un ensemble de rectangles, quelques milliers, qui est ensuite arrangé dans une texture d'*atlas* par un algorithme *glouton*. Les résultats de cette méthode dépendent fortement du seuil de coefficients non-nuls fixé pour la segmentation des données. Un compromis doit être trouvé entre la densité de chaque région et le nombre de rectangles à gérer dans la texture

d'atlas. L'espace perdu par l'arrangement dans la texture d'atlas est environ de 3%.

Pour encoder une *BRDF*, Claustres *et al.* [CBP07] proposent une méthode basée sur la suppression pour obtenir une représentation compressée à base d'ondelettes d'une *BRDF*. L'idée réside dans l'expression d'une *BRDF* 4D dans une texture 3D. Chaque texture 2D composant la texture 3D correspond à une direction de point de vue où chaque pixel représente une direction de lumière. Les textures de points de vue sont individuellement décomposées par la suite dans une base d'ondelettes de Haar. La compression est réalisée en supprimant les informations pouvant être retrouvées par le principe de réciprocité. Ces informations redondantes sont donc considérées comme une texture nulle. Un index est créé pour identifier les textures (i.e. directions sortantes) présentes ou nulles.

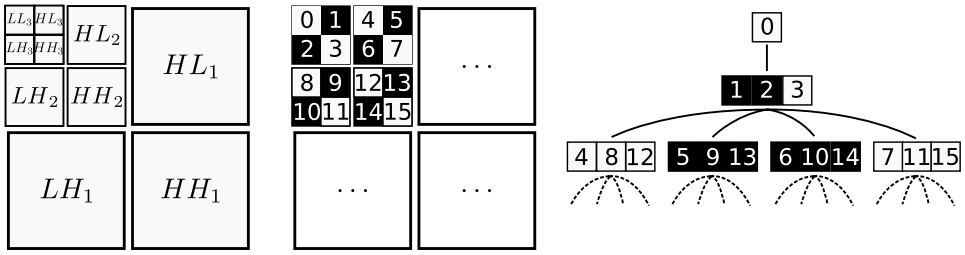


FIGURE 3.7: Arbre de coefficients défini par Boulton [Bou08].

Dans [Bou08], Mike Boulton présente une méthode générique pour compresser une décomposition en ondelettes d'une image et l'utiliser sur le matériel graphique dans de nombreux domaines d'application. Cette approche consiste à créer un arbre quaternaire de coefficients tel que décrit sur la Figure 3.7. Les coefficients sont regroupés par paquets de trois coefficients qui représentent un noeud de l'arbre. Chaque noeud a quatre fils qui sont les détails nécessaires à la reconstruction du prochain niveau de résolution. Cette arborescence a déjà été présentée dans la littérature [BBS94] pour l'édition multi-résolution d'images et permet d'encapsuler en un seul noeud toutes les informations nécessaires à la reconstruction d'une résolution plus fine pour un pixel donné. Le codage proposé ici consiste à stocker l'arbre de décomposition dans une texture 1D et à munir chaque noeud d'une valeur de décalage (i.e. *offset*) qui indique le saut à effectuer jusqu'au premier fils. Il s'agit donc d'un codage d'arbres quaternaires complets où un noeud possède quatre fils ou aucun. Cette représentation exploite le fait que l'arbre de décomposition est creux sur les feuilles et donc, permet d'éliminer un maximum d'informations inutiles (i.e. coefficients nuls) lors d'une seuillage des coefficients. La valeur du décalage stockée peut être rapidement grande si l'arbre de décomposition est profond. Pour utiliser un décalage codé sur peu de bits, l'image d'origine est découpée en sous-images auxquelles un arbre de décomposition est associé individuellement.

Les méthodes présentées ici sont un moyen d'exploiter une décomposition en ondelettes sur le matériel graphique actuel. Néanmoins, les taux de compression comparativement à la méthode du codage *Zero-Tree* sont bien en deçà. Un compromis plus efficace est nécessaire entre le ZTC et les méthodes typées matériel pour permettre une utilisation plus large des ondelettes sur les processeurs graphiques programmables.

Deuxième partie

Modèles et représentations proposés

Chapitre 4

Ondelettes et *BTF* : Premières expérimentations

En informatique graphique, les ondelettes sont le plus souvent utilisées sur des domaines réguliers en deux dimensions telles que des images ou des textures. La popularité de cette utilisation vient d'une part, de la facilité de mise en oeuvre de tels outils et d'autre part, de l'efficacité d'une base d'ondelettes *simple* pour la mise en place de différentes échelles de détail. Dans un contexte de compression, les facultés de décorrélation d'une base d'ondelettes sont exploitées de manière à représenter une texture avec un nombre restreint de coefficients. La projection d'une *BTF* dans une base linéaire a déjà été expérimentée dans l'état de l'art (voir Section 2.2). Dans les approches antérieures, l'espace propre associée à une *BTF* reste la base privilégiée pour réduire la dimensionnalité. Cependant, aucune approche ne propose une projection dans une base multi-échelle ce qui limite l'utilisation des méthodes existantes tant pour le rendu sur matériel graphique qu'en terme de scalabilité.

Ici, la *BTF* sera considérée suivant les deux approches privilégiées de l'état de l'art : un ensemble de textures bi-directionnelles ou un ensemble de *ABRDF* par pixel. Sur chacune des configurations, une décomposition en ondelettes en 2D est appliquée (Section 4.1) pour mettre en avant les facultés d'une telle représentation suivant une transformation simple. Les avantages statistiques de la décomposition en ondelettes sur les *BTFs* se démontrent dans la pratique lorsque cette représentation est utilisée dans un codage de type *Zero-Tree* (Section 4.2). Ces résultats seront analysés et comparés avec les méthodes de l'état de l'art pour dégager des pistes vers de nouvelles approches (Section 4.3).

4.1 Mise en place d'une *BTF* multi-échelle

De nombreuses configurations sont envisageables pour exprimer une *BTF*

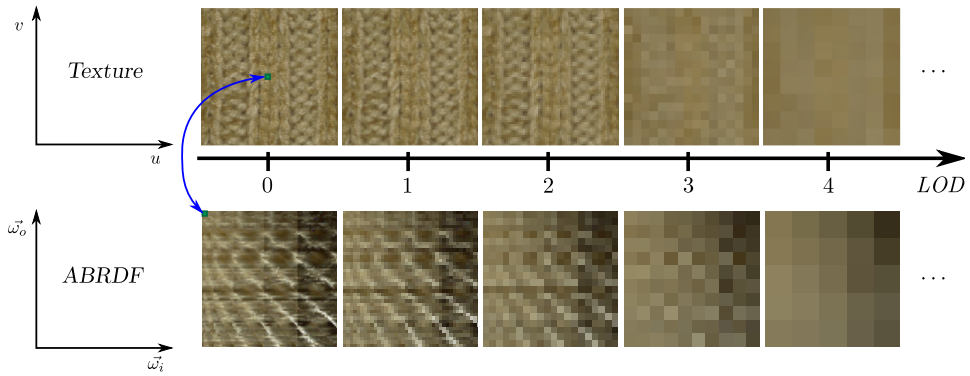


FIGURE 4.1: Exemple de niveaux de détail (*LOD*) sur une texture de *BTF*. En haut, une décomposition en ondelettes 2D est appliquée sur une texture directionnelle. En bas, une texture représentant une *ABRDF* est projetée dans une base d'ondelettes 2D. *Note : pour des raisons de lisibilité, les échelles de chaque texture ne sont pas respectées.*

dans une base d'ondelettes multi-résolution. Pour limiter le nombre de configurations possibles, une transformation 2D seulement est appliquée sur l'ensemble d'images qui composent une *BTF*. La similarité inter-image n'est donc pas exploitée ici. Nous utilisons uniquement les ondelettes de première génération pour dégager les avantages du regroupement des deux domaines de recherche : ondelettes et *BTF*.

Suivant l'approche choisie, chaque image représente soit la variation spatiale d'un matériau pour une direction de vue et une direction de lumière données, soit la variation angulaire d'un échantillon spatial du matériau, représentée par une *ABRDF*. Ces deux possibilités sont présentées sur la Figure 4.1. La formalisation de cette transformation consiste à exprimer la *BTF* comme une fonction de sept dimensions :

$$BTF\left(x, \vec{\omega}_o, \vec{\omega}_i\right) = BTF\left(x, \vec{\omega}_o, \vec{\omega}_i, \sigma\right), \quad \text{avec } \sigma = 0$$

où σ représente l'échelle de détail. Par cohérence avec les notations en programmation graphique, le niveau de détail $\sigma = 0$ correspond au niveau de détail le plus fin.

Les deux configurations possibles amènent à analyser la décomposition en ondelettes suivant deux approches différentes. Dans le cas où on considère la *BTF* comme un ensemble de textures, où chaque texture représente une information directionnelle unique, la définition ensembliste est :

$$\left\{ T_{(\vec{\omega}_o, \vec{\omega}_i)}(x, \sigma) \mid \vec{\omega}_o \in \mathbb{W}_o,\ \vec{\omega}_i \in \mathbb{W}_i \right\}, x \in \mathbb{T}.$$

Dans cette approche, le lien avec les méthodes issues de l'analyse d'images par transformation en ondelettes est directe : un groupe de quatre pixels

voisins au niveau de détail j est approximé par un seul pixel au niveau de détail $j+1$. La représentation multi-échelle telle que définie ici représente une bonne approximation des *véritables* niveaux de détail des dimensions spatiales. En effet, les dimensions spatiales de la *BTF* sont à la fois une représentation de la méso-structure et de la réflectance du matériau. Par conséquent, les échelles de détail spatiales nécessitent la mise en place de méthodes multi-échelles géométriques pour définir la méso-structure sous différentes échelles. Néanmoins, le calcul de la méso-géométrie n'est pas fiable avec les données issues de l'acquisition actuelles à cause de problèmes de recalage ou d'alignement de perspectives. Une approche basée image reste donc la plus sure.

Dans le cas où la *BTF* est un ensemble de *ABRDFs*, pour laquelle chaque échantillon spatial est associé à une *ABRDF*, la notation multi-échelle de cette approche est :

$$\{ABRDF_x(\omega_o, \omega_i, \sigma)\ ,\ x \in \mathbb{T}\}\,,\ \omega_o \in \eta_o \,\text{et}\, \omega_i \in \eta_i.$$

L'analyse de cette configuration est beaucoup plus ambigüe. En effet, ici, un pixel d'une texture de *ABRDF* représente une direction. Ainsi, entre deux niveaux de détail, un groupe de quatre pixels (i.e. directions) d'une texture au niveau j est approximé par un seul pixel (i.e. direction) au niveau $j+1$. Or, dans une *ABRDF*, les dimensions angulaires sont linéarisées pour former une texture en 2D, les deux dimensions représentent les directions du point de vue et de la lumière. La projection des dimensions angulaires dans un espace de plus faible dimension fait que deux directions voisines dans $\mathbb{S}^2$ ne sont pas forcément voisines dans l'espace 1D linéarisé. L'approximation des échelles de détail d'une telle texture ne permet donc pas de décrire, pour un niveau de détail σ donné, quelle direction est représentée par un pixel. De plus, cette observation entraîne une perte de cohérence *spatiale* dans la texture *ABRDF* et limite les facultés de décorrélation d'une transformée en ondelettes. Néanmoins, dans cette étude expérimentale, cette configuration est retenue pour envisager l'ensemble des possibilités.

4.2 Codage *Zero-Tree*

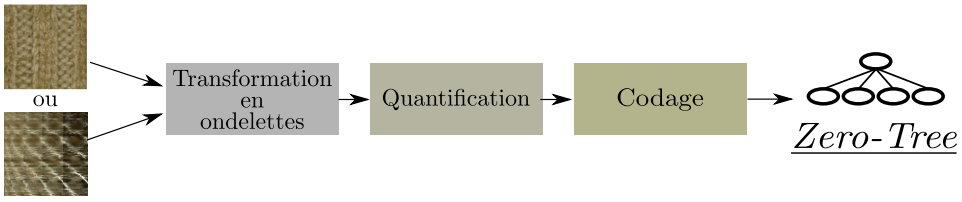


FIGURE 4.2: Schéma de codage développé pour compresser les textures d'une *BTF*.

Le codage mis en place ici est une version épurée du codage JPEG2000 [JPE00], un schéma « classique » en compression d'images à base d'ondelettes. Comme présenté sur la Figure 4.3, le processus se déroule en trois étapes consécutives :

1. Décomposition en ondelettes ;
2. Quantification ;
3. Codage en *Zero-Tree*.

Le dispositif accepte comme données d'entrée une texture qui dans notre cas peut être une texture directionnelle ou une texture *ABRDF*. Le format de sortie est un flux de bits représentant le codage en *Zero-Tree*.

4.2.1 Décomposition en ondelettes

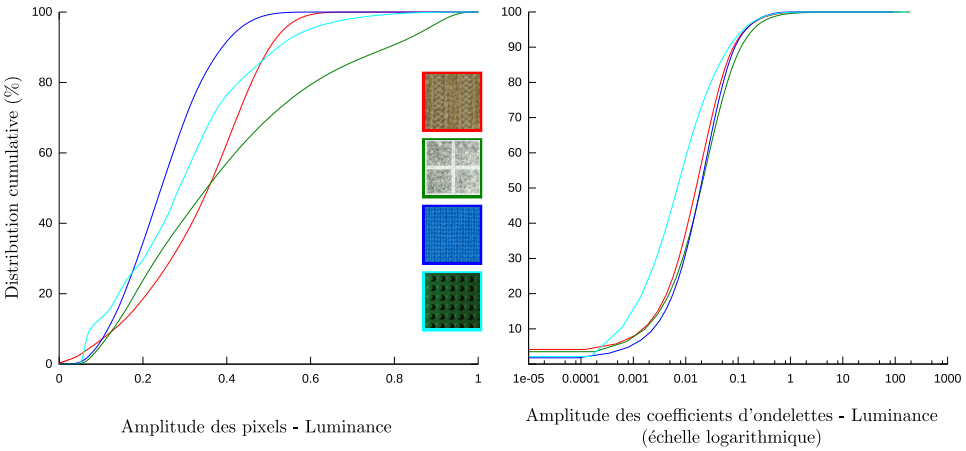


FIGURE 4.3: Distribution cumulative de différents matériaux pour toutes les textures de la vue normale uniquement. A gauche, la courbe représente la distribution des valeurs de luminance des textures d'origine. A droite, la distribution des mêmes coefficients projetés dans une base d'ondelettes de Daubechies (D4).

Pour la compression d'images, une transformée en ondelettes est particulièrement performante grâce à ses facultés de décorrélation. En effet, comme présentée sur la Figure 4.3, la distribution des coefficients avant et après décomposition change radicalement. Sur ces courbes, l'axe des ordonnées représente la distribution cumulative de textures issues de différentes *BTFs* par rapport à l'amplitude des coefficients, notée en abscisse. Ici, toutes les textures pour le point de vue normal ont été prises en compte et chacune, transformée individuellement à l'aide d'une décomposition en ondelettes de Daubechies (D4), la base d'ondelettes retenue dans notre schéma de codage. Ainsi, on peut remarquer que, dans le cas général, plus de 90% des coefficients sont en dessous de la valeur 0.01 et ce pour chaque matériau exprimé

dans une base d'ondelettes. A l'inverse, en considérant uniquement la luminance de chaque pixel sans transformation (à gauche), la distribution est fortement dépendante du jeu de données et difficilement prédictible.

L'analyse précise de la distribution cumulative des coefficients d'ondelettes d'une BTF permet de connaître la nature d'un matériau. En effet, même si l'allure globale des courbes de la Figure 4.3 dans l'espace ondelettes est relativement proche, on peut distinguer un décrochage pour certains matériaux, notamment le matériau « lego » dont la courbe croît plus rapidement que les autres. Cette observation vient du fait que ce matériau comporte relativement peu de haute-fréquences, représentées par des coefficients de faible magnitude dans une transformée en ondelettes, contrairement au matériau « laine » dont les mailles engendrent une plus grande quantité de haute-fréquences. Les basses fréquences, coefficients de forte amplitude dans l'espace ondelettes, sont relativement peu nombreux mais sont indispensables dans ce type de représentation.

Une méthode de compression à base d'ondelettes consiste à seuiller les coefficients (voir Section 3.3). Cette approche révèle tout son potentiel lorsqu'on observe la distribution des coefficients mais peut s'avérer dangereuse car un compromis subtil doit être trouvé entre la taille des données et la qualité de la représentation. Ici, le seuillage des coefficients est réalisé durant l'étape de quantification.

4.2.2 Quantification

Le type de quantification retenu est identique à celui du format JPEG2000 [JPE00] appelée *quantification avec zone morte* [1]. Comme illustrée sur la Figure 4.4, la notion de zone morte fait référence au fait que la zone de quantification autour de 0 est différente des autres intervalles de quantification. Dans le cas général, la fonction de quantification associée est définie telle que :

$$\begin{cases} Q(x) &= \left\lfloor \frac{x+(1-f).\Delta}{\Delta} \right\rfloor = q \\ R(q) &= q.\Delta \approx x \end{cases}$$

où Q est la fonction de quantification d'un réel x et R la fonction de reconstruction d'une valeur quantifiée q avec un pas de quantification Δ. Le paramètre $f \in [0; \frac{1}{2}]$ permet de définir la zone morte. Si $f = \frac{1}{2}$ alors la quantification est une quantification uniforme ; si $f = 0$ alors la taille de l'intervalle de quantification autour de 0 est deux fois plus importante que celles des autres intervalles. La taille de la zone morte définit l'importance donnée aux coefficients de très faible amplitude dans la transformée en ondelettes.

Dans l'application présentée ici, la valeur du paramètre f est fixée à $\frac{1}{3}$, empiriquement testée comme paramètre de quantification quasi sans perte. La valeur absolue des coefficients d'ondelettes est codée sur 8 bits non-signés

1. Traduction littérale de Dead-Zone Quantization.

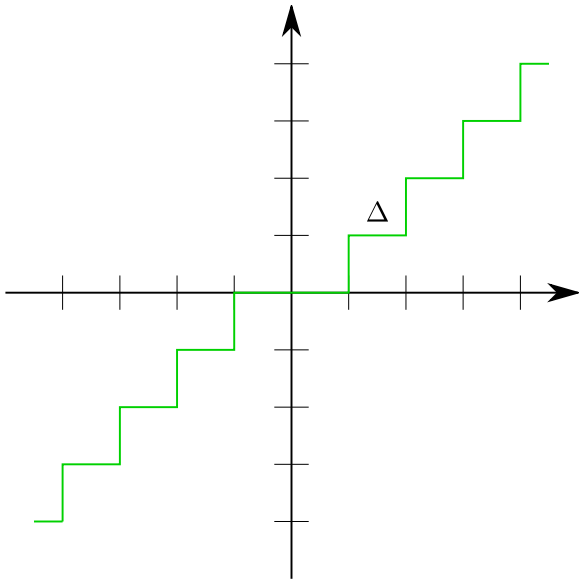


FIGURE 4.4: Quantification avec zone morte. L'intervalle de quantification autour de 0 est deux fois plus grand que les autres intervalles. Δ représente le pas de quantification.

et le signe de chaque coefficient est stocké directement lors du codage *Zero-Tree*. La valeur du pas de quantification Δ est calculée en prenant l'amplitude maximale des coefficients d'ondelettes divisée par 2^8.

4.2.3 Codage

Parmi les méthodes basées *Zero-Tree* de l'état de l'art (voir Section 3.3), l'algorithme utilisé ici est *SPIHT* [SP96], une amélioration de la méthode originelle du codage *Zero-Tree* [Sha93]. Cette méthode correspond a un codage entropique de complexité réduite permettant un codage progressif, en qualité ou en résolution, d'un arbre issu d'une décomposition en ondelettes. De plus, l'avantage de *SPIHT* est de pouvoir être utilisé sans codeur arithmétique, contrairement à [Sha93], sans perte significative de performance de compression mais par contre, avec une complexité d'exécution relativement faible.

Le codeur *SPIHT* est un algorithme itératif où à chaque itération correspond un plan de bits, en partant du bit de poids le plus fort vers le bit de poids le plus faible. Cette approche permet soit de quantifier à la volée les coefficients d'ondelettes en considérant les réels générés par une transformée en ondelettes comme des réels à virgule fixe soit, comme dans notre méthode, de quantifier au préalable les coefficients et d'encoder directement les valeurs quantifiées. Une itération se déroule en deux étapes : une étape de *tri* et une étape de *raffinement*.

L'étape de tri consiste à déterminer quels sont les coefficients *significatifs* pour le plan de bit courant. Un coefficient est dit *significatif* si son bit de poids fort est sur le même plan de bits que celui de l'itération courante. Cette information est transmise au codeur en plus de l'information du signe et ce, dans un ordre particulier pour connaître les coordonnées du coefficient sans les stocker. La deuxième étape consiste à raffiner les coefficients jugés comme significatifs aux itérations précédentes (i.e. les coefficients pris en compte dans l'étape de tri pour les plans de bits précédents). Ainsi dans cette étape, l'algorithme transmet, dans un ordre connu du codeur, les bits du plan courant des coefficients significatifs des itérations précédentes.

L'algorithme itère de nouveau à l'étape de tri en se plaçant sur le plan de bits suivant jusqu'à atteindre le plan de bits de poids le plus faible. Dans le schéma de codage présenté ici, les coefficients sont stockés sur 8 bits. La méthode itère donc en premier lieu sur le plan de bits 7, bit de poids le plus fort, et itère jusqu'au plan de bits 0, bit de poids le plus faible. L'ordre dans lequel les coefficients sont parcourus influence les performances d'un tel codage. Dans *SPIHT*, le parcours des coefficients est effectué des basses fréquences vers les hautes fréquences pour présenter en premier les coefficients ayant le plus de chance de devenir significatif. L'ordre mineur de parcours est ensuite géré par un ensemble de listes permettant de partitionner les coefficients dans un agencement optimal pour un parcours minimisant le nombre de bits nécessaires au codage.

SPIHT est une méthode de codage sans perte ce qui implique que les erreurs d'approximation dans l'image reconstruite ne sont pas dues au codage mais aux transformations précédentes : transformée en ondelettes et quantification. L'avantage d'un tel format réside dans l'aspect progressif[2] du flux de bits généré par l'algorithme. En effet, une fois le signal encodé, le flux de bits peut être interrompu n'importe où si un critère qualitatif ou quantitatif est atteint. Cette particularité est exploitée dans la partie résultat (4.3).

4.3 Résultats et discussion

Les résultats du schéma de codage proposé sont présentés sur le Tableau 4.1. L'approche retenue pour obtenir ces résultats est d'utiliser notre schéma de codage comme une compression sans perte en gardant l'intégralité du flux de bits généré par l'algorithme. Ici, la méthode est qualifiée comme sans perte si le *PSNR*, métrique permettant de juger la qualité d'une image avec référence, est supérieure à 50 dB. Dans le cas général, les différences entre l'image reconstruite et l'image d'origine avec un tel *PSNR* sont imperceptibles par l'oeil humain. Deux configurations ont été testées pour compresser une *BTF* : la configuration textures directionnelles et celle textures d'*ABRDF* (voir Section 4.1).

2. « *embedded code* » dans la langue de Shakespeare.

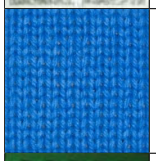
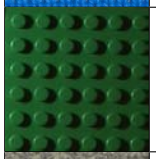
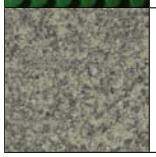
Matériaux	Dimensions			Original (Mo)	ZeroTree (Mo)		Ratio (%)
	η_x	η_i	η_o		*ABRDF*	*Tex*	
	256 × 256	81	81	1228	633	**243**	20
	256 × 256	81	81	1228	611	**272**	22
	256 × 256	81	81	1228	742	**322**	26
	192 × 192	120	90	1126	677	**204**	18
	800 × 800	81	81	11980	*nc*	**2457**	20

TABLE 4.1: Compression sans perte réalisée à l'aide du schéma de codage proposé. Les matériaux utilisés ici sont issus de [BTFa] et [KMBK03]. Les deux configurations de codage, par *ABRDF* ou par texture, sont exposées. Le ratio détermine le pourcentage de la taille des données compressées par rapport à la taille du matériau original. Le codage est réalisé suivant les trois composantes colorimétriques.

Les résultats montrent que pour un tel schéma, la compression spatiale est beaucoup plus efficace que la compression angulaire. Cette observation est le fruit de plusieurs liens de causalité. En effet, les textures d'*ABRDF* ont généralement une résolution moins importante que les textures spatiales et, de par l'expérience tirée de l'utilisation de méthodes de compression à base d'ondelettes dans de nombreux domaines graphiques, ce type de schéma fonctionne d'autant mieux que la résolution des textures est grande. D'autre part, la cohérence des données est moins importante dans le domaine angulaire linéarisé que dans le domaine spatial, ce qui explique la différence de taille entre les deux configurations. En observant le ratio maximum de compression pour chaque matériau, obtenu systématiquement dans la configuration textures directionnelles, cette approche se révèle très efficace car

les taux de compression varient de 4 à 5 suivant le matériau considéré. A cause des défauts induits par le flux de bits, ce codage ne permet pas l'exploitation de ce format de données pour un rendu en temps réel sur matériel graphique mais, par contre, le justifie comme une méthode de stockage efficace des *BTFs*.

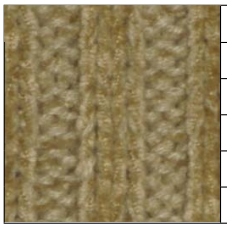
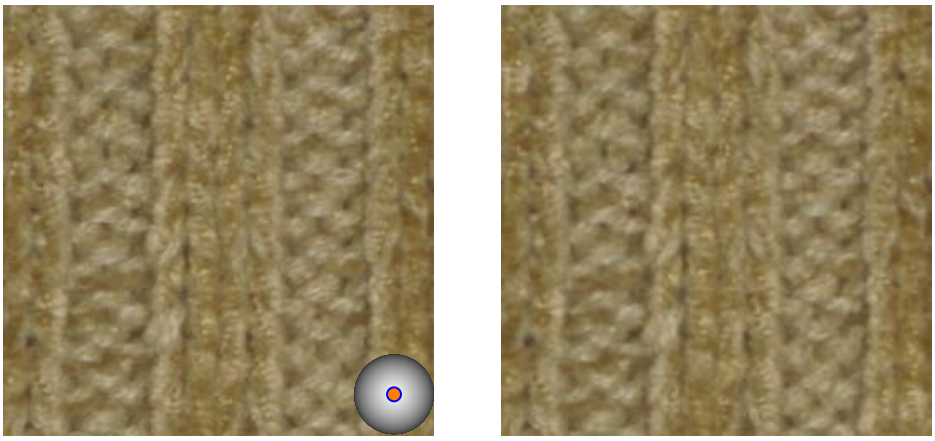
	Bits par pixel	Taille (Mo)	PSNR moyen (dB)
	24 (original)	1228	∞
	4,74 (sans perte)	243	∞
	1	51	38,90
	0,5	25,3	34,70
	0,1	5,4	28,50

TABLE 4.2: Le matériau Pulli [BTFa] compressé à l'aide de la méthode du *ZeroTree Coding*. Ici, le nombre de bits par pixel varie pour apprécier les variations de qualité de compression.

Vu que l'on dispose d'un flux de bits progressif représentant une BTF, le flux peut être partiellement exploité pour correspondre à un critère qualitatif ou quantitatif. Le Tableau 4.2 expose les résultats obtenus en ne prenant en compte qu'une partie du flux de bits, pour chaque texture, dont la taille est fixée en fonction du nombre de bits par pixel (bpp) que l'on souhaite obtenir. La BTF originale représentant le matériau *Pulli* correspond à un codage de 24 bpp : 8 pour chaque composante RVB. Pour un codage sans perte réalisé avec notre méthode, le nombre de bits par pixel est de 4,74. L'idée ici est de juger de la qualité du matériau pour un nombre de bpp bien inférieur au format d'origine. On peut remarquer ainsi que, en considérant le $PSNR$ comme une métrique perspicace, les résultats montrent un $PSNR$ moyen satisfaisant pour un codage supérieure à 0,5 bpp. En observant les résultats visuels de cette méthode de compression extrême (voir Figure 4.5), les effets perceptuels commencent à devenir gênants en deçà de 1 bpp. Ces résultats sont particulièrement probants car la taille d'une BTF, 51 Mo pour le cas critique, par rapport à la qualité de la reconstruction, $PSNR$ moyen égal à 38,90 dB, est un très bon compromis si on se réfère aux méthodes de l'état de l'art (voir Section 2.2).

Cet algorithme a été valorisé par la mise en place d'un brevet international [BG08a]. Dans ce document, la méthode est généralisée à un processus en N dimensions. Lorsque $N \leq 4$, la BTF est considérée comme une fonction $2D$, représentant les textures spatiales de la BTF, qui varie suivant les dimensions angulaires : linéarisation des points de vue et des directions de la lumière si $N = 3$ ou linéarisation indépendante de chaque dimension angulaire si $N = 4$. Une autre configuration consiste à fixer $N = 6$ et, le cas échéant, à décomposer les dimensions angulaires à l'aide d'ondelettes géométriques. Prendre en compte les dimensions directionnelles pour la com-



(a) 24 bpp (original)

(b) 1 bpp



(c) 0,5 bpp



(d) 0,1 bpp

FIGURE 4.5: Le matériau « Pulli » [BTFa] compressé à l'aide de la méthode du codage *Zero Tree*. Les différences commencent à être visibles en deçà de 1 bit par pixel. La texture reconstruite correspond au point de vue $\vec{\omega}_o = (0, 0)$ et à la direction de lumière $\vec{\omega}_i = (0, 0)$.

pression permet d'envisager de meilleurs taux de compression. En effet, bien que non probant dans notre approche du fait de la projection de l'espace des directions dans un espace 2D, les échantillons angulaires sont aussi fortement corrélés. Pour un même point de vue et différentes directions de la lumière, de nombreuse méthodes de l'état de l'art exploitent la redondance d'informations de ces échantillons. Suivant les directions du point de vue, les effets de parallaxe limitent la corrélation entre les échantillons mais, suivant les variations de la méso-structure du matériau, il est aussi envisageable de prendre en compte ces dimensions pour la compression.

Les résultats présentés ici sont satisfaisants en terme de compression mais souffrent d'une limitation forte pour une techniques de compression « moderne » : la portabilité sur matériel graphique. En effet, le parallélisme de masse des cartes graphiques actuelles est un bon moyen d'accélérer les calculs et, plus particulièrement en informatique graphique, de visualiser les données de manière efficace. Cet avantage n'est pas seulement exploité dans les méthodes en temps réel mais aussi dans les méthodes de rendu dites hors ligne, qui utilise ce paradigme pour accélérer les calculs. Cependant, cette approche est restrictive car tout processus ne peut être porté directement sur ce type d'architecture s'il ne respecte pas certaines règles, entre autres :

- parallélisme
- peu ou pas de branchements conditionnels
- format et type de données restrictifs
- cohérence de cache

Le format de données présenté ici ne respecte pas les critères de mise en oeuvre principalement pour des raisons de structure de données. En effet, chaque texture est représentée par un flux de bits. Le type atomique sur matériel graphique est le type entier, au minimum sur 8 bits, ce qui nous permet de stocker le flux de bits comme un tableau d'entiers sur 8 bits par concaténation des données par paquet de 8 bits. Les opérations bit à bit sont disponibles depuis peu en programmation graphique et permettent donc de manipuler le flux. Cependant, ces opérations sont coûteuses et pour un pixel donné lors du rendu, l'accès aux informations du pixel n'est pas dynamique : un lourd processus de décodage nécessitant des listes de stockage et des analyses bit à bit est nécessaire. Par conséquent, il n'est pas (encore) envisageable de porter ce type de méthode sur matériel graphique. Dans nos approches futures, la structure de données développée doit prendre en compte cet état de fait au risque de souffrir des mêmes limitations. Pour cela, nous proposons dans le chapitre suivant une solution dans ce sens où le codage par flux de bits est abandonné au profit d'un codage par bloc de données.

Chapitre 5

Arbre d'ondelettes polynômial

Lors du chapitre précédent, nous avons étudié la mise en place d'une transformée en ondelettes sur les *BTFs*. Ces travaux ont montré que les propriétés de décorrélation d'une décomposition en ondelettes permettent de mettre en place des méthodes de compression efficaces sur ce type de matériau. Le schéma de codage présenté précédemment ne permet pas une visualisation en temps réel d'une *BTF* via les processeurs graphiques programmables. Dans ce chapitre, une représentation progressive d'une *BTF* est développée. Son principe repose sur une approximation polynomiale des données suivie d'une transformation en paquet d'ondelettes. L'agencement particulier des coefficients d'ondelettes par ce type de transformation est ensuite exploitée pour permettre un rendu en temps réel sur le matériel graphique.

5.1 Présentation

L'*arbre d'ondelettes polynômial* est une structure de données qui, comme son nom le laisse supposer, est basée sur l'approximation polynomiale en conjonction avec les ondelettes. Le but d'une telle méthode est de générer représentation compacte de fonctions spatio-directionnelles utilisables directement sur le matériel graphique. L'exemple applicatif est réalisé ici sur les *BTFs* mais cette approche n'est pas limitée à ce type de matériau et peut être utilisée sur des *SVBRDFs* sans difficulté. Cependant, l'utilisation sur les *BTFs* reste plus intéressant car elles peuvent représenter une plus grande classe de matériaux, notamment grâce aux effets de parallaxe.

Le schéma global de cette méthode est présenté sur le Figure 5.1. La transformation d'une *BTF* se déroule en plusieurs étapes. Premièrement, chaque point de vue est traité indépendemment et les textures directionnelles représentant les variations de direction lumineuses sont approximées par pixel à l'aide d'un polynôme bivarié. Cette première étape constitue en soit une méthode de compression car le nombre de coefficients polynomi-

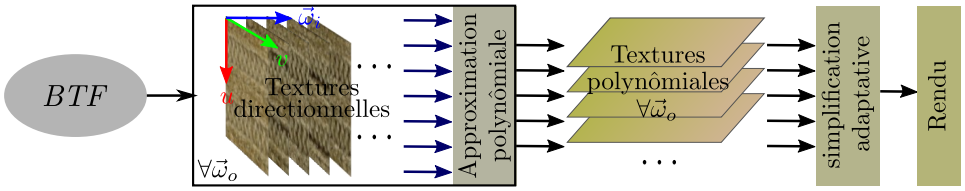


FIGURE 5.1: Schéma général de la méthode proposée. Le dispositif transforme une *BTF* en représentation compacte pour le rendu en deux étapes : une approximation polynomiale de textures directionnelles et une simplification adaptative des textures polynomiales.

aux est bien inférieur au nombre de directions de la lumières. Néanmoins, de fortes redondances d'informations, notamment spatiales, demeurent dans cette représentation. Pour éliminer les informations superflues, l'ensemble des textures polynomiales, une par point de vue, est analysé en exprimant chaque texture en sous-bandes fréquentielles. Chaque sous-bande représente un apport d'énergie dans la représentation et, en ordonnant les sous-bandes par ordre croissant d'énergie, une représentation progressive des textures polynomiales peut être mise en place. Cette simplification adaptative permet de générer des représentations très compactes pour une *BTF* et autorise donc un rendu en temps réel sur matériel graphique.

5.2 Approximation polynomiale

5.2.1 Choix de la base d'approximation

Initialement proposée pour un point de vue fixe [MGW01], l'approximation polynomiale a été généralisée pour un ensemble de points de vue sur les *BTFs* [MMK04a]. Ce type d'approximation est réalisé à l'aide de polynômes bivariés qui décrivent le comportement d'un pixel pour un point de vue fixe et toutes les directions de la lumière. Pour la plupart des matériaux, cette approche est la plus raisonnable car les transitions suivant l'espace des directions de la lumière sont plus douces que celles entre les différents points de vue. En effet, la méso-géométrie engendre des effets de parallaxe qui entraîne une perte de cohérence de la *BTF* sur ces dimensions. Un polynôme de degré faible peut donc être utilisé ce qui simplifie la complexité des calculs d'approximation et améliore les taux de compression.

Dans la littérature, on retrouve deux cas de figure principaux pour l'approximation de *BTFs* : la factorisation par un modèle de *BRDF* analytique et l'approximation polynomiale (cf. Section 2.2). Déterminer quelle approche est la plus noble dépend du type de matériau à traiter. En effet, pour des matériaux à méso-structure planaire, la factorisation par un mod-

èle de *BRDF* est une approche très efficace et se retrouve dans la plupart des approximations de *SVBRDFs*. Lorsque la méso-structure du matériau perturbe le comportement directionnel (ombre ou parallaxe), les modèles de *BRDF* utilisables sont bien trop limités pour représenter tous les phénomènes physiques. En revanche, l'approximation polynomiale se comporte relativement bien par rapport aux effets méso-structurels. La principale limitation est l'approximation des effets haute-fréquences où les pics spéculaires sont généralement émoussés du fait du degré et du type de polynôme approximant qui n'autorise que des transitions douces. Cependant, d'après nos expérimentations, l'approximation polynomiale reste la méthode la plus adaptée pour représenter une grande classe de matériaux.

Dans les méthodes précédentes [MMK04a], l'approximation polynomiale sur une *BTF* est définie par point de vue, $\vec{\omega}_o$, comme :

$$\begin{aligned} BTF_{\vec{\omega}_o}(x, \vec{\omega}_i) &\approx P_{\vec{\omega}_o}(x, \Pi(\vec{\omega}_i)) \\ &= \sum_{i,j} a_{i,j}(x) p_u^i p_v^j. \end{aligned} \tag{5.1}$$

où $\Pi(\vec{\omega}_i)$ est la projection de la direction incidente de la lumière dans l'espace 2D de la texture $(\theta_l, \phi_l) \mapsto (p_u, p_v)$ et $a_{i,j}(x)$ sont les coefficients polynômiaux définis par pixel x. Les degrés de liberté disponibles pour ce type d'approximation sont donc :

- le choix du vecteur de projection,
- le type de projection utilisé,
- le degré du polynôme.

Comme spécifié dans l'équation 5.2, le vecteur de projection utilisé dans l'état de l'art est le vecteur $\vec{\omega}_i$ représentant la direction incidente de la lumière. La fonctionnelle permettant d'exprimer le vecteur $\vec{\omega}_i$ dans $\mathbb{R}^2$ est une projection orthographique dont la transformation est :

$$\begin{aligned} \mathbb{S}^2 &\rightarrow \mathbb{R}^2 \\ (\theta_l, \phi_l) &\mapsto (sin(\theta_l)cos(\phi_l), sin(\theta_l)sin(\phi_l)) \end{aligned}$$

Enfin, le degré du polynôme bivarié est 2, représenté seulement par six coefficients, tel que :

$$P(p_u, p_v) = a_0 p_u^2 + a_1 p_v^2 + a_2 p_u p_v + a_3 p_u + a_4 p_v + a_5.$$

Bien que relativement efficaces, les outils présentés dans la littérature ne sont pas optimaux car ils ne reflètent pas les spécificités d'une *BTF*. Par exemple, l'espace de paramétrisation orthographique est sur-défini autour de la normale alors que les variations lumineuses les plus importantes sont localisées sur les angles rasants. Ces particularités doivent être prises en compte dans cette méthode de paramétrisation pour minimiser les erreurs d'approximation.

5.2.2 Paramétrisation de l'espace des directions

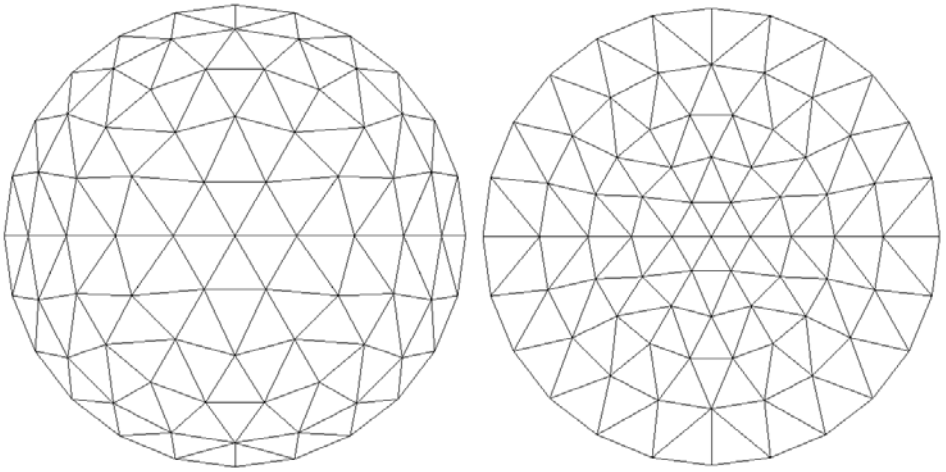


FIGURE 5.2: Projection orthographique (à gauche) versus projection paraboloïde (à droite) des directions de la lumière (format triangulé) issu de [BTFa]. Les angles rasants dans l'espace paraboloïde ont un échantillonage supérieur par rapport à la projection orthographique.

Dans la plupart des matériaux disponibles issus de *BTFs*, les différences dans l'espace des directions de la lumière sont principalement dues :

- aux propriétés du matériau lui-même : paramètres de réflexion, facteur d'atténuation inclus dans la *BTF*, diffusion sous-surfacique ;
- à la méso-structure : ombres portées, éclairage indirect, direction de la normale en chaque point.

Des études sur le comportement lumineux de différents matériaux ont montré que l'intensité de la réflexion augmente à mesure que l'angle devient rasant [NDM05], caractérisé par l'effet de *Fresnel*, présent dans de nombreuses *BTFs*. Un espace de paramétrisation constant suivant les points de vue est donc à proscrire et l'adaptativité de cet espace suivant les points de vue doit être définie. Pour une *BTF*, l'orientation des normales à la surface n'est pas connue et l'espace texture est considéré comme le repère local au matériau pour mesurer les angles de vues et de lumières. En prenant en compte toutes ces observations, nous proposons l'utilisation d'un espace de paramétrisation plus efficace pour minimiser les erreurs d'approximation polynomiales.

Privilégier les angles rasants dans l'espace de projection de la lumière implique que l'angle solide doit diminuer à mesure que l'angle zénithal augmente. Heidrich *et al.* [HS98] ont proposé une paramétrisation de l'espace des directions, appelée projection paraboloïde, qui répond à ces attentes. Dans

cette approche, l'image capturée par une caméra orthographique à travers une paraboloïde, définie telle que

$$f(x, y) = 0.5 + 0.5(x^2 + y^2), x^2 + y^2 \leq 1,$$

contient toutes les informations de l'hémisphère frontal. Comme illustré sur la Figure 5.2, dans une paramétrisation paraboloïde, les directions perpendiculaires à la direction normale sont échantillonnées plus finement que les directions parallèles au point de vue. Comme démontré par le calcul dans la partie 5.2.3, cet espace de projection est plus opportun pour l'approximation polynomiale car le polynôme tend vers les extrema, localisés sur les angles rasants, d'une manière plus douce.

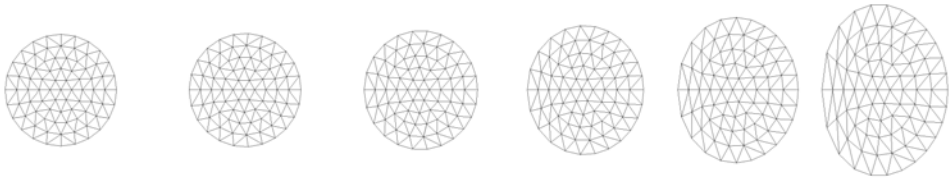


FIGURE 5.3: Projection paraboloïde du vecteur $\vec{H}$ pour des points de vue différents. De gauche vers la droite, $\vec{\omega}_o$ est égal à : $(0, 0)$, $(15, 0)$, $(30, 0)$, $(45, 0)$, $(60, 0)$ et $(75, 0)$. Pour chaque point de vue, les échantillons de la lumière, triangulés, sont projetés dans l'espace paraboloïde à l'aide du vecteur $\vec{H}$.

Cependant, ce changement de paramétrisation ne permet pas une distinction entre les points de vue. Or, il est préférable d'adapter la paramétrisation suivant le point de vue qu'on souhaite approximer, notamment sur les points de vue rasant dont l'intensité de réflexion est plus importante suivant les directions opposées de la lumière. Pour se faire, nous choisissons d'introduire dans la représentation le vecteur bissecteur $\vec{H}$, défini par Rusinkiewicz [Rus98], formulé tel que :

$$\vec{H} = \frac{\vec{\omega}_o + \vec{\omega}_i}{\|\vec{\omega}_o + \vec{\omega}_i\|}.$$

Pour chaque point de vue, l'espace de paramétrisation des directions de la lumière est défini comme la projection paraboloïde du vecteur bissecteur. Comme illustré à la Figure 5.3, l'espace de paramétrisation généré ainsi est adapté à chaque point de vue : l'échantillonnage des directions de la lumière augmente à mesure que l'angle entre la direction du point de vue et la direction de la lumière augmente. Cet espace permet de mieux représenter la réflectivité de certains matériaux suivant les angles rasants du point de vue. Les résultats chiffrés sont présentés dans la partie 5.2.3.

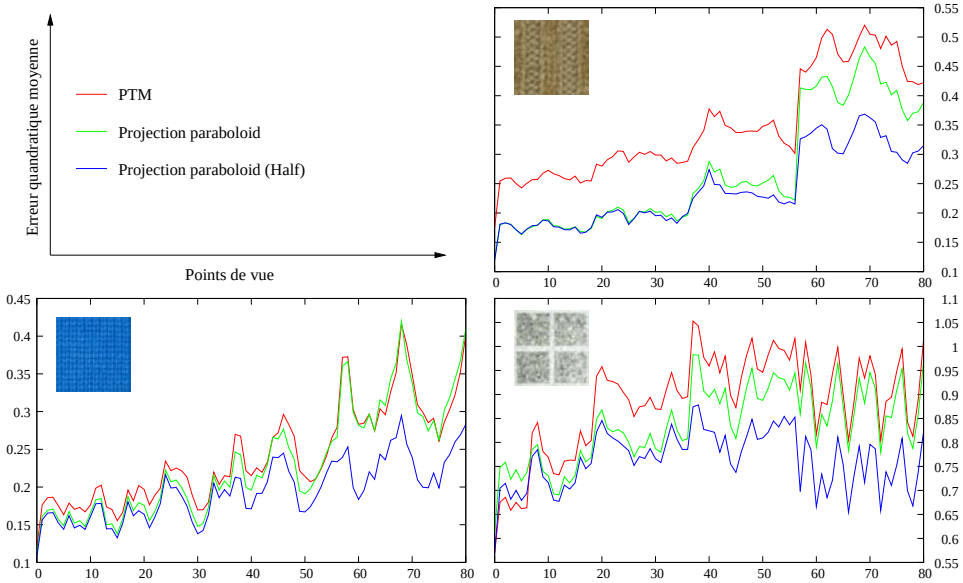


FIGURE 5.4: Comparaison de l'erreur d'approximation par point de vue obtenue en fonction du type de projection utilisé pour l'espace des directions de la lumière. Les résultats qualitatifs par rapport à la méthode originale (PTM [MGW01]) montrent que l'espace de paramétrisation que nous proposons est plus adapté à l'approximation polynomiale.

5.2.3 Résultats sur l'erreur d'approximation

Les résultats présentés ici ont été effectués par approximation polynomiale, avec des polynômes bivariés de degré 2, sur la luminance suivant différentes projection de l'espace des directions. La métrique employée pour estimer l'erreur d'approximation est définie par point de vue telle que :

$$e(\vec{\omega}_o) = \frac{1}{\eta_x} \sum_{\vec{\omega}_i} \sum_{x} (BTF_{\vec{\omega}_o}(x, \vec{\omega}_i) - P_{\vec{\omega}_o}(x, \vec{\omega}_i))^2, \tag{5.2}$$

c'est à dire, la somme des erreurs quadratiques de toutes les directions de la lumière pour un pixel moyen.

La Figure 5.4 montre l'erreur d'approximation pour trois matériaux différents en fonction du point du vue. Dans le cas général, les résultat obtenus en utilisant la projection paraboloïde du vecteur $\vec{\omega}_i$ minimisent les erreurs d'approximation de 5 à 40% en fonction des matériaux et des points de vue. En adaptant la paramétrisation au point de vue, les erreurs d'approximation sont encore diminuées, principalement sur les points de vue rasants (points de vue indexés par une valeur supérieure à 55 sur l'axe des abscisses) où la différence de qualité d'approximation reste visible mais minimisée. Un exemple visuel est illustré sur la Figure 5.5 représentant une texture du matériau

« pulli » pour un angle rasant. Cette différence qualitative entre les points de vue rasants et les autres observations n'est pas due uniquement aux erreurs d'approximation. En effet, pour tous ces points de vue, l'étape de rectification des textures lors de l'acquisition (cf. Section 2.1) est particulièrement sensible. De ce fait, pour un même point de vue, la correspondance inter-pixel suivant les directions de la lumière n'est pas forcément respectée et fausse donc les résultats d'approximation.

Certaines observations marginales montrent que notre paramétrisation n'est pas efficace dans certains cas. Pour le matériau « impalla », représenté sur la Figure 5.4 en bas à droite, les premiers points de vue sont approximés plus efficacement à l'aide d'une projection orthographique des directions de la lumière. La distribution des normales à la surface pour ce type de matériau, composé de cristaux, est complexe et par conséquent, les propriétés de réflectivité sont plus difficiles à prédire. Un espace de paramétrisation constant n'est pas forcément à éviter dans ce type de configuration. Néanmoins, les résultats restent globalement meilleurs en utilisant la projection orthographique de vecteur bissecteur.

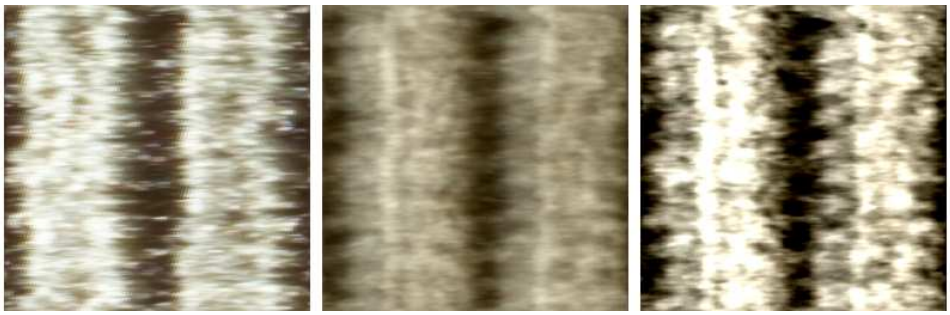


FIGURE 5.5: A gauche, la texture originale pour $\vec{\omega}_o = (75, 0)$ et $\vec{\omega}_i = (75, 180)$; au milieu, texture reconstruite par la méthode de [MMK04a] ; à droite, texture reconstruite en utilisant notre paramétrisation.

La taille de cette représentation pour les matériaux issus de [BTFa] où la résolution spatiale est $\eta_x = 256 \times 256$ pour $\eta_o = 81$ points de vue et $\eta_i = 81$ directions de la lumière est :

$$\eta_x * \eta_o * 6 * 3 = 95\,551\,488 \text{ coefficients}$$

où le multiplicateur 6 correspond au nombre de coefficients polynômiaux par pixel et 3, le nombre de composantes colorimétiques (RVB). La taille en mega-octet est donc de 120 Mo pour chaque composante où les coefficients polynômiaux sont représentés chacun par un réel sur 32 bits. Le taux de compression est d'environ 50% en considérant chaque pixel d'une *BTF* originale comme une donnée codée sur 24 bits. L'avantage d'une telle approximation,

en plus de la compression, est la continuité de l'espace des directions de la lumière qui est plus efficace qu'une interpolation entre les échantillons lors du rendu. Cependant, la taille de la représentation est encore trop importante pour répondre aux critères de rendu en temps réel sur matériel graphique. Une méthode de compression plus efficace est donc proposée par la suite sur ce type de données afin de transformer la représentation polynomiale en structure permettant d'adapter la taille de la représentation suivant des critères définis.

5.3 Simplification adaptative

Dans cette section, nous introduisons une méthode de simplification adaptative de signaux 2D. L'approche retenue ici est de découper une texture en sous-bandes fréquentielles de taille fixe. A chaque sous-bande est associée un poids caractérisant l'apport de cette sous-bande dans la représentation. Un ordre dans les sous-bandes peut donc être mis en place en triant les sous-bandes par ordre d'importance et ainsi, suivant un critère quantitatif ou qualitatif, garder les n premières sous-bandes pour représenter le signal.

Cet algorithme de simplification se déroule en deux étapes. La première consiste à transformer un signal 2D en un ensemble de sous-bandes fréquentielles. Pour se faire, la méthode des paquets d'ondelettes, décrite dans la partie 5.3.1, est utilisée pour projeter les données dans un espace d'analyse permettant de qualifier l'importance de chaque sous-bande dans la représentation. La deuxième étape décrit la simplification adaptative en classifiant les sous-bandes, construites par la transformation en paquets d'ondelettes, par ordre d'importance puis en rejetant des sous-bandes qui ne respectent pas un critère adaptatif donné (cf. partie 5.3.2).

5.3.1 Paquets d'ondelettes polynomiaux

Comme décrite dans le Chapitre 4, une décomposition en ondelettes est un bon moyen d'extraire le minimum d'informations nécessaire pour représenter un signal. Nos expérimentations ont démontré que l'utilisation des ondelettes sur les dimensions spatiales d'une *BTF* était efficace car la corrélation des données y est très forte. Le problème lors de l'utilisation d'une telle transformation n'est pas son efficacité mais comment stocker judicieusement les coefficients résultants de cette transformation. Les méthodes de codage basées *Zero-Tree* sont, certes, les plus adaptées mais ne permettent pas de visualiser le flux généré directement sur matériel graphique.

Pour limiter le coût de stockage de telles données, nous avons choisi une approche par bloc à travers une décomposition en paquets d'ondelettes sur les dimensions spatiales des textures polynomiales. Comme présentée à la Figure 5.6, une transformation en paquets d'ondelettes consiste à décomposer récursivement toutes les bandes de fréquence. Après un seul niveau de

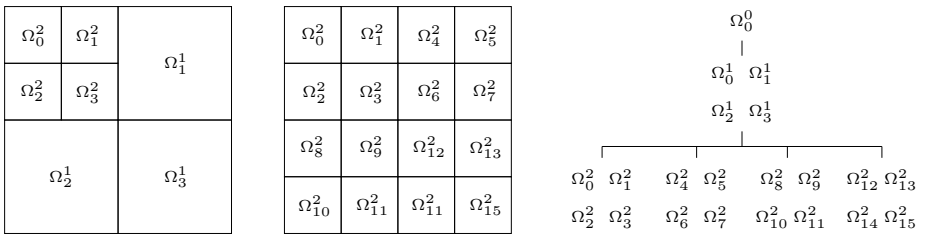


FIGURE 5.6: Une décomposition en ondelettes « classique » (à gauche) versus une décomposition en paquets d'ondelettes (au milieu) après deux niveaux de décomposition. Une transformation en paquets d'ondelettes décomposent récursivement toutes les bandes de fréquences alors qu'une décomposition en ondelettes « classique » transforme récursivement la sous-bande d'approximation uniquement, $\Omega_0^j \ \forall j$. L'arbre de sous-bandes généré par la décomposition en paquets d'ondelettes est illustré à droite.

décomposition, la transformation est identique à l'approche classique et sépare le signal en quatre sous-bandes : une sous-bande d'approximation Ω_0^1 et trois sous-bandes de détails Ω_1^1, Ω_2^1 et Ω_3^1. Pour les niveaux de décomposition suivant, l'ensemble des sous-bandes est décomposé récursivement, y compris les sous-bandes de détails.

Cette approche est souvent utilisée pour déterminer la base d'ondelettes idéale suivant un certain critère. Une fonction de coût est appliquée à une sous-bande Ω_ι^j pour un niveau de décomposition j. Après transformation en ondelettes de la sous-bande Ω_ι^j, la somme des coûts des sous-bandes résultantes, $\Omega_{4\iota}^j$, $\Omega_{4\iota+1}^j$, $\Omega_{4\iota+2}^j$ et $\Omega_{4\iota+3}^j$, est calculée. Si cette somme est inférieure au coût initiale, coût de la sous-bande Ω_ι^j, alors la décomposition est appliquée sinon la décomposition est arrêtée au niveau j. Par exemple, une fonction de coût peut être construite comme le nombre de coefficients d'une sous-bande inférieurs à un seuil. Auquel cas, la base d'ondelettes générée par cette analyse est celle qui maximise le nombre de coefficients inférieur à un seuil. Ici, la notion de base adaptative n'est pas la qualité recherchée. Si le même niveau de décomposition j est appliqué à l'ensemble des sous-bandes, on obtient une structure régulière de 4^j sous-bandes de même taille. Cette uniformité dans la structure de chaque sous-bande nous permet d'envisager des structure de données efficaces pour le stockage sur matériel graphique.

Dans la Section 5.2, l'approximation polynomiale d'une BTF est représentée par un ensemble de textures polynomiales où chaque texture correspond à un point de vue. Un pixel d'une telle représentation est donc un vecteur de n coefficients, où n varie suivant le degré de polynôme utilisé. L'analyse en paquets d'ondelettes peut donc être appliquée sur ces textures de manière analogue à une transformation de texture au format RVB. Chaque texture polynomiale est projetée dans le domaine spatio-fréquentiel des ondelettes

en utilisant une décomposition par paquets. Ainsi, une sous-bande Ω_ι^j est un ensemble de coefficients d'ondelettes polynômiaux où chaque pixel est un vecteur de n dimensions représentant les n coefficients polynômiaux dans l'espace ondelettes. L'indice j indique le niveau de décomposition et ι est un indice spatial identifiant une région spatiale de la texture. Dans cet esprit, l'indice ι peut être défini sur l'ensemble des sous-bandes polynomiales indépendemment du point de vue correspondant et dans ce cas, $0 \leq \iota < \eta_o * 4^j$ pour un niveau de décomposition j.

En considérant une sous-bande Ω_ι^j comme un sous-ensemble spatial dépendant de la vue, l'ensemble des polynômes $\mathbb{P}_\iota^j$ d'une sous-bande peut être noté comme :

$$\mathbb{P}_\iota^j = \{P_{\vec{\omega}_o}(x, \vec{\omega}_i) | x \in \Omega_\iota^j, \iota \rightsquigarrow \vec{\omega}_o\}.$$

pour un niveau de décomposition $j > 0$, un indice ι, et où $\iota \rightsquigarrow \vec{\omega}_o$ signifie que cet indice est associé au point de vue $\vec{\omega}_o$. Les points de vue peuvent être indexés par un identifiant unique ω_o tel que $\vec{\omega}_o \rightarrow \omega_o$, avec $0 \leq \omega_o < \eta_o$, auquel cas une sous-bande Ω_ι^j est associée au point de vue d'indice $\omega_o = \left\lfloor \frac{\iota}{4^j} \right\rfloor$. Chaque sous-ensemble polynômial $\mathbb{P}_\iota^j$ a une contribution dans la représentation globale de la BTF polynomiale que nous allons quantifier.

5.3.2 Prédiction d'erreur

Une hiérarchie intrinsèque de l'amplitude des coefficients est donnée par la transformation en paquets d'ondelettes : les amplitudes fortes sont localisées dans les sous-bandes basse-fréquences et les sous-bandes haute-fréquences sont moins significatives. Cette observation amène à penser qu'un sous-ensemble de sous-bandes suffit à exprimer le signal sans impact visuel gênant. Pour pondérer la contribution de chaque sous-bande $\mathbb{P}_\iota^j$ dans la représentation, nous définissons une fonction d'énergie, E, telle que :

$$E(\mathbb{P}_\iota^j) = \sum_{x \in \Omega_\iota^j} \sum_{\vec{\omega}_i \in \mathbb{W}_i} \left(P_{\vec{\omega}_o}(x, \vec{\omega}_i)\right)^2,$$

qui représente la contribution énergétique d'une sous-bande polynomiale $\mathbb{P}_\iota^j$ dans la représentation. Ainsi, on définit un ordre dans les sous-bandes à l'aide d'une permutation π telle que :

$$\mathbb{P}_{\pi(0)}^j, \mathbb{P}_{\pi(1)}^j, \cdots, \mathbb{P}_{\pi(\eta_o * 4^j - 1)}^j$$

avec,

$$E(\mathbb{P}_{\pi(0)}^j) \leq E(\mathbb{P}_{\pi(1)}^j) \leq \cdots \leq E(\mathbb{P}_{\pi(\eta_o * 4^j - 1)}^j)$$

Cette ordonnancement permet de considérer alors la représentation polynomiale de la BTF issue de la transformation en paquets d'ondelettes comme une représentation progressive.

En effet, la fonction $E(\mathbb{P}_\iota^j)$ est directement liée à l'erreur quadratique engendrée par la suppression de la sous-bande polynomiale $\mathbb{P}_\iota^j$. Considérons le signal R comme la BTF reconstruite par l'ensemble des $\eta_o * 4^j$ sous-bandes de la représentation et $\tilde{R}$ la reconstruction de la BTF avec seulement s sous-bandes, $s < \eta_o * 4^j$. Le carré de l'erreur L^2 de cette reconstruction est :

$$\|R - \tilde{R}\|_2^2 = \sum_{k=0}^{\eta_o * 4^j - s} E(\mathbb{P}_\iota^j). \tag{5.3}$$

Ce résultat nous permet de prédire la quantité d'erreur introduite dans la représentation entre la reconstruction complète R, avec toutes les sous-bandes polynomiales, et une reconstruction partielle $\tilde{R}$, à partir des s sous-bandes de plus fortes pondérations. Notre métrique est donc basée sur l'erreur quadratique introduite dans la représentation par la suppression d'une sous-bande mais toute fonction d'estimation de la qualité peut être utilisée ici. L'avantage avec l'erreur quadratique est son utilisation sans référence contrairement à des métriques plus couramment utilisées, tel que le PSNR, qui nécessitent une comparaison avec les données d'origine.

Tous les points de vue d'une BTF n'ont pas la même importance : les points de vue proches de la vue frontale sont plus importants, visuellement parlant, que ceux dans les angles de vue rasants. En effet, les angles de vue rasants sont sous-échantillonnés lors du rendu lorsque l'on utilise des niveaux de détails. La probabilité pour qu'un des ces points de vue soient visualisés au niveau de détail le plus fin est faible. Par conséquent, la métrique E est pondérée par un poids en fonction de l'angle d'élévation du point de vue de la sous-bande polynomiale traitée :

$$EV(\mathbb{P}_\iota^j) = E(\mathbb{P}_\iota^j) * \cos(\theta_o).$$

où θ_o représente l'angle d'élévation du point de vue associé à $\mathbb{P}_\iota^j$.

Grâce à cette représentation progressive, on peut modifier la taille ou la qualité de l'approximation suivant des critères quantitatifs ou qualitatifs. La mise en place de critères qualitatifs consiste à normaliser l'énergie pondérée de chaque sous-bande polynomiale par le facteur

$$\sum_{\iota=0}^{\iota<\eta_o*4^j} EV(\mathbb{P}_\iota^j)$$

et ainsi, de définir un pourcentage de l'énergie totale à conserver dans l'approximation progressive. Pour les critères quantitatifs, l'algorithme consiste à éliminer les sous-bandes de plus faible énergie jusqu'à une taille de représentation voulue.

5.3.3 Résultats de la simplification adaptative

Les résultats présentés ici sont réalisés en approximant la luminance des différents matériaux avec des polynômes bivariés de différents degrés. Sur cette approximation, la simplification adaptative est appliquée suivant des critères de taille ou de qualité comme décrit précédemment. La finesse de la simplification est guidée par le nombre de décompositions durant l'analyse en ondelettes. Ici, pour tous les matériaux, le niveau de décomposition est égale à 6. Ainsi, chaque sous-bande a une résolution spatiale de 8×8. Les résultats sont présentés dans les Figures 5.1, 5.2 et 5.3 sur des matériaux différents. Pour chacun des résultats illustrés, un tableau décrit les caractéristiques choisies lors de l'approximation et la simplification. Une courbe d'erreur suivant les points de vue estime la qualité du modèle par rapport à la *BTF* d'origine à l'aide d'une fonction d'erreur, identique à celle formulée dans l'équation 5.2.

En plus des résultats quantitatifs et qualitatifs de la simplification adaptative, nous avons choisi ici, au vu de la taille des représentations après simplification, de changer le degré du polynôme approximant pour tester de nouveaux résultats d'approximation. Le degré du polynôme bivarié utilisé est spécifié dans la colonne *Polynôme* :

- Q : polynômes de degré 2 ;
- Q/C : polynômes cubiques et quadratiques.

Dans l'approche libellée Q/C, les sous-bandes les plus significatives, la moitié, sont approximées par des polynômes bivariés de degré 3, représentés chacun par 10 coefficients réels, alors que l'autre moitié est approximée par des polynômes de degré 2 seulement. Le critère guidant la simplification est identifiée dans la colonne *Critère* où soit la taille de la représentation est imposée soit un pourcentage de l'énergie totale. Pour chaque configuration proposée, la taille de la représentation ainsi que le pourcentage d'énergie conservé sont spécifiés. Enfin, pour tous les matériaux, une configuration de référence AOP0, où AOP signifie *arbre d'ondelettes polynômial*, est donnée à titre illustratif et correspond à la reconstruction d'une approximation polynomiale de degré 2 sans simplification.

Le résultats présentés ici montrent qu'une grande partie de la représentation peut être exclue du signal tout en gardant la courbe représentant l'erreur d'approximation tangente à celle de la configuration de référence. De ce fait, les résultats quantitatifs sont très performants et le meilleur compromis entre la taille et la qualité permet de représenter une *BTF* par une structure d'environ 30 Mo. Cependant, des taux de compression plus importants peuvent aussi être appliqués au risque de voir apparaître lors du rendu des effets de blocs (cf. Figure 5.7 a).

Bien que diminuant fortement les erreurs d'approximation, l'utilisation de polynômes cubiques n'est pas visuellement pertinente. En effet, l'augmentation du degré du polynôme n'est plus valable pour un degré supérieur à

Wool	Taille (Mo)	E	Polynôme	Critère
AOP0	120	1.0	Q	-
AOP1	**30**	0.72	Q	taille
AOP2	60	0.90	Q/C	énergie
AOP3	35	0.72	Q/C	taille

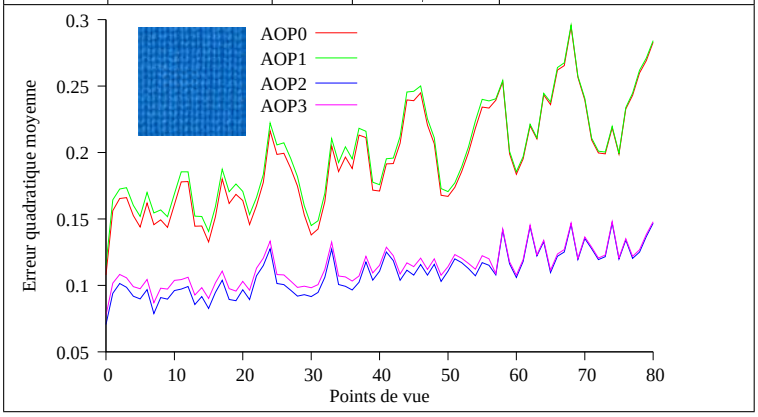


TABLE 5.1: Résultats sur le matériau « wool » issu de l'université de Bonn [BTFa].

Pulli	Taille (Mo)	E	Polynôme	Critère
AOP0	120	1.0	Q	-
AOP1	**4**	0.50	Q	énergie
AOP2	12	0.63	Q	taille
AOP3	35	0.80	Q/C	taille

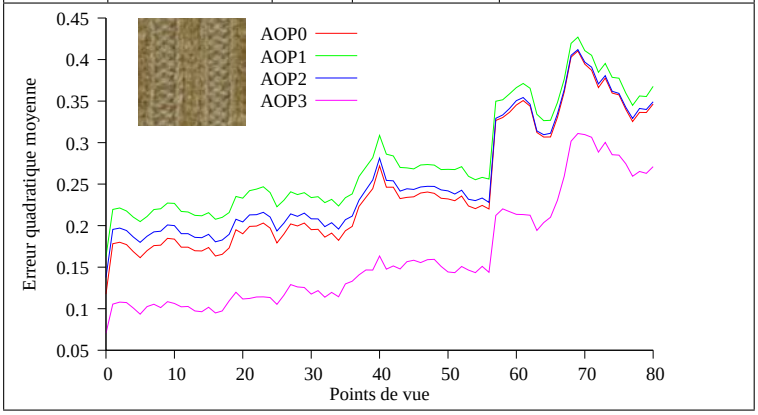


TABLE 5.2: Résultats sur le matériau « pulli » issu de l'université de Bonn [BTFa].

Impalla	Taille (Mo)	E	Polynôme	Critère
AOP0	120	1.0	Q	-
AOP1	**30**	0.81	Q	taille
AOP2	37	0.90	Q/C	énergie

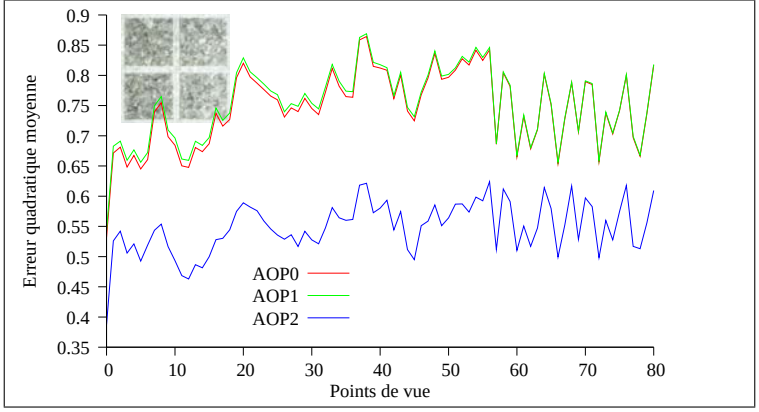
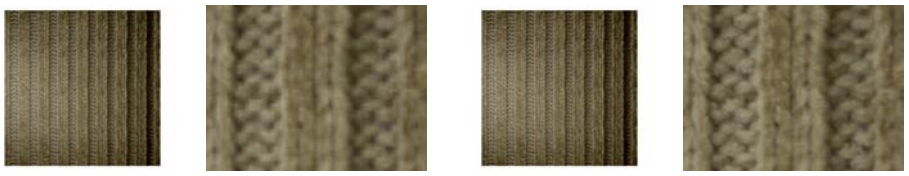


TABLE 5.3: Résultats sur le matériau « impalla » issu de l'université de Bonn [BTFa].

3. Lors de nos expérimentations, nous avons utilisé des polynômes d'ordre supérieur qui minimisent un peu plus les erreurs d'approximation mais qui oscillent trop pour obtenir une représentation lisse de l'espace des directions de la lumière.



(a) (b)

FIGURE 5.7: Résultats visuels de configurations présentées sur la Figure 5.2. (a) AOP Quadratique avec E=0.5 de 4Mo, à gauche, image reconstruite. (b) AOP Quadratique/Cubique avec E=0.8 de 35Mo. Pour (a) et (b) sont présentés, à gauche, une image reconstruite et à droite, un zoom sur une partie de l'image reconstruite.

5.4 Rendu

Une fois la représentation polynomiale adaptative générée, les sous-bandes polynomiales peuvent être stockées en mémoire vidéo pour un rendu en temps réel sur matériel graphique. Cette opération nécessite un agencement par-

ticulier des données pour disposer efficacement de la représentation sur les processeurs graphiques programmables, les *shaders*. Ce type de programmation autorise uniquement des structures de données régulières de type texture en deux ou trois dimensions. Pour cela, les sous-bandes sont stockées par groupes pour minimiser le nombre de textures en mémoire vidéo.

Supposons que nous disposons d'une représentation polynomiale adaptative telle que décrit précédemment où le niveau de décomposition est fixé à 2. Chaque texture polynomiale, une par point de vue, est donc un ensemble de 4^2 sous-bandes polynomiales, soit $\eta_o * 4^2$ pour tous les points de vue. Après simplification adaptative, on dispose de n sous-bandes pour représenter le signal. A cette étape, on alloue en mémoire vidéo un tableau de texture 2D suffisant pour stocker l'ensemble des sous-bandes dont les dimensions spatiales varient en fonction de la taille de chaque sous-bande. En effet, allouer un tableau de taille n, dont la résolution spatiale de chaque texture égale à celle d'une sous-bande, n'est pas efficace car la résolution d'une sous-bande est relativement petite, 8×8 ou 16×16. Dans le cas général, pour une BTF dont la résolution spatiale est $\eta_x = \eta_u \times \eta_v$, la résolution de chaque texture vidéo est fixée à $\frac{\eta_u}{2} \times \frac{\eta_v}{2}$. La profondeur du tableau est ajustée pour pouvoir stocker les n sous-bandes.

S_0^2	S_1^2	S_4^2	$\not{S}_5^2$
S_2^2	S_3^2	$\not{S}_6^2$	S_7^2
S_8^2	S_9^2	S_{12}^2	$\not{S}_{13}^2$
$\not{S}_{10}^2$	S_{11}^2	S_{14}^2	$\not{S}_{15}^2$

Ω_0^2	Ω_1^2	Ω_4^2	Ω_5^2
Ω_2^2	Ω_3^2	Ω_6^2	Ω_7^2
Ω_8^2	Ω_9^2	Ω_{12}^2	Ω_{13}^2
Ω_{10}^2	Ω_{11}^2	Ω_{14}^2	Ω_{15}^2

FIGURE 5.8: Carte des sous-bandes significatives. Si une sous-bande Ω_ι^j n'est pas exclue de la représentation, un index S_ι^j est stockée pour identifier ses coordonnées dans la texture de sous-bandes polynomiales. Dans le cas contraire, un index nul est affecté, $\not{S}_\iota^j$, pour spécifier que la sous-bande est exclue de la représentation.

A chaque point de vue, on associe une carte de sous-bandes significatives. Comme spécifié sur la Figure 5.8, cette carte permet d'identifier les sous-bandes polynomiales présentes dans la représentation simplifiée. Un index, représenté par un entier sur 16 bits, est attribué à chaque sous-bande à condition qu'elle soit présente dans la représentation et localise la place de la sous-bande dans la texture contenant l'ensemble des sous-bandes polynomiales. Cette index est initialisé à zéro si la sous-bande est exclue de la représentation.

L'arbre d'ondelettes polynômial est utilisé pour le rendu uniquement sur les valeurs de la luminance de la BTF. La BTF est donc au préalable ex-

primée au format de couleur YUV où Y représente la luminance et UV les informations chromatiques. Le traitement des deux composantes chromatiques est plus simple car les variations de ces dernières suivant les directions de la lumière sont faibles. Ainsi, une texture chromatique est utilisée par point de vue représentant la moyenne par pixel des composantes UV suivant les directions de la lumières. Les η_o textures résultantes sont ensuite quantifiée de manière uniforme sur des entiers de 8 bits. Ces informations constituent en mémoire un taille d'environ 10 Mo.

L'algorithme de reconstruction des données est réalisé sur un *shader* qui agit par pixel sur l'image de synthèse. Cette reconstruction peut être réalisée en une seule passe sur le matériel graphique actuel. A chaque pixel, on associe un niveau de détail spécifiant à quelle résolution l'échantillon de l'approximation de BTF doit être reconstruit. A partir de l'estimation du point de vue et de la direction de la lumière dans un repère local, les coefficients d'ondelettes sont reconstruits puis combinés, suivant la base utilisée, pour synthétiser la valeur du pixel au niveau de détail demandé. L'interpolation suivant les points de vue est réalisée par le calcul des coordonnées barycentriques du point de vue à reconstruire à partir de la triangulation de l'espace des directions du point de vue. Le filtrage spatial est tri-linéaire car interpolé suivant les niveaux de détails et les dimensions spatiales.

Dans cette approche, le nombre d'images par seconde est dépendant du niveau de décomposition utilisé lors de la transformation en paquets d'ondelettes. Le rendu est en temps réel, nombre d'images par seconde supérieur à 60, pour un niveau de décomposition inférieur ou égale à 6. Pour des niveaux de décomposition supérieurs, le rendu est interactif voire en temps réel si le nombre de sous-bandes composant la représentation est faible.

5.5 Discussion et approches futures

Les arbres d'ondelettes polynômiaux représentent une alternative intéressante aux méthodes de l'état de l'art. D'une part, les taux de compression sont adaptatifs et on remarque que pour des compressions extrêmes, de l'ordre de quelques mega-octets, la qualité de la représentation reste correcte. Cette observation est due à la pondération de la métrique d'erreur qui permet de garder les informations sur les points de vue proches de la vue normale jusqu'à de très forts taux de compression. D'autre part, cette approche permet d'effectuer un rendu en temps réel d'une BTF directement sur le matériel graphique.

Ces travaux ont été valorisés à travers un brevet [BG08b] ainsi qu'une présentation orale [BBGS08] à la conférence SIGGRAPH en 2008. Dans ces publications, des approches expérimentales ont été étudiées pour approximer le résiduel entre une BTF et son approximation polynomiale telle que définie dans la Section 5.2. L'approximation du résiduel est un moyen d'a-

grémenter la représentation polynomiale par un ajout de singularités suivant certaines directions de la lumière qui participent au réalisme d'un matériau synthétique. Le résiduel représente les erreurs d'approximation d'une fonction quadratique (ici, le polynôme) et, de ce fait, ne peut pas être approximé avec des modèles analytiques de degré inférieur ou égal à 2. L'approche retenue est donc de modéliser cette différence géométriquement par une surface en 2.5D : une surface 2D, dans l'espace paramétrique formé par Π, munie d'une élévation en chaque point, le résiduel.

Pour limiter la taille des données, un maillage adaptatif ne peut être choisi pour chaque pixel. De plus, le nombre de points du maillage doit être restreint pour ne pas surcharger la représentation. Un maillage fixe pour l'ensemble des pixels est donc proposé avec un nombre de points (i.e. directions) égal à 13, comme présenté à la Figure 5.9, et les directions intermédiaires sont calculées par interpolation barycentrique. Un degré de liberté supplémentaire est proposé pour le maillage approximant par l'ajout d'une phase $\alpha \in [0; \frac{\pi}{2}[$, rotation du maillage autour du centre. Cette phase permet d'aligner les axes du maillage contenant le plus d'informations avec les zones où le résiduel est maximal. L'approximation du résiduel par le maillage approximant revient donc à trouver α et les valeurs du résiduel en chaque point tels que l'erreur entre les valeurs du résiduel d'origine et son approximation soit minimale au sens des moindres carrés.

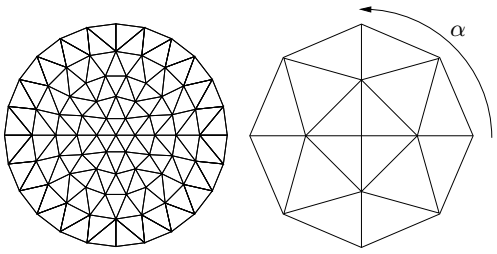


FIGURE 5.9: Maillage pour le résiduel géométrique : à gauche, maillage issu de l'acquisition ; à droite, maillage d'approximation.

Les expérimentations que nous avons effectuées à partir de cette approche ont été réalisées à l'aide d'un système multi-résolution sous contrainte. En effet, calculer α indépendemment sur chaque pixel peut provoquer des artefacts visuels lors du rendu sur les zones homogènes du matériau : la phase entre deux pixels voisins doit être proche. Des contraintes sont donc posées sur α telles que la différence en deux phases voisines soit inférieure à un seuil.

Lors des résultats expérimentaux, l'erreur d'approximation est améliorée d'environ 10%. Le gain sur la qualité de l'approximation par rapport au nombre de coefficients ajoutés n'est donc pas celui escompté. Cette analyse nous a tout de même permis de mettre en avant les défauts de l'approximation polynomiale. En effet, sur la Figure 5.10, l'erreur relative sur l'approxima-

tion polynomiale, telle que décrite dans la Section 5.2, est présentée suivant les niveaux de détails d'une décomposition en ondelettes au niveau 4 pour chaque point de vue d'une *BTF*. Ce tracé montre que les basses fréquences (niveau de composition 4) sont approximées pratiquement sans erreur. En revanche, plus les niveaux de détails sont fins (niveau de composition 3, 2 et 1) plus l'erreur relative est importante.

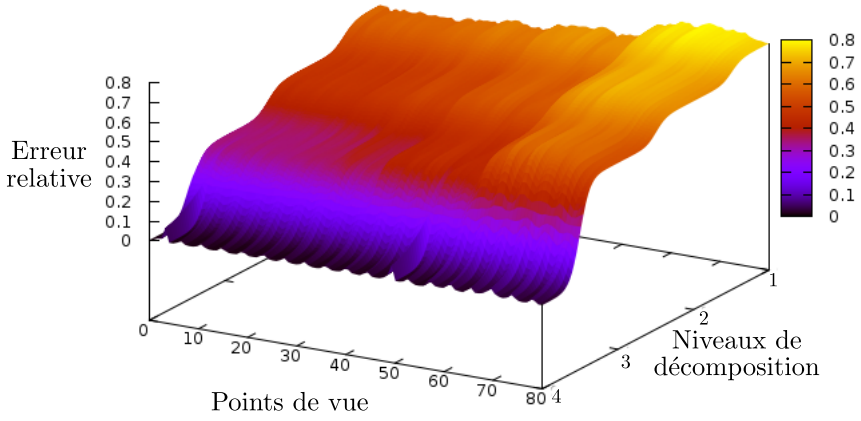


FIGURE 5.10: Erreur relative de l'approximation polynômiale sur une *BTF* suivant les points de vue et les niveaux de décomposition en ondelettes. L'approximation polynômiale est robuste sur les coefficients d'approximation (niveau 0) mais ne permet pas de représenter les haute-fréquences (niveau > 0).

Ces observations nous laissent envisager des possibilités pour des approches futures à base d'approximations mixtes. En effet, comme présenté sur la Figure 5.10, les erreurs d'approximation ne sont pas les mêmes suivant les niveaux décomposition en ondelettes. Pour le niveau d'approximation (niveau 0), l'erreur relative d'approximation est quasi-nulle et plus les niveaux de décomposition sont fins, plus l'erreur augmente. Cette constatation démontre que les polynômes sont limités pour représenter les haute-fréquences d'une *BTF*. Ainsi, l'approximation polynômiale pourrait être réalisée uniquement sur les coefficients d'approximation et les détails, encodés d'une manière différente. Dans le chapitre suivant, nous développons une méthode capable de stocker les hautes fréquences de manière efficace. Une approche mixte à partir de ces deux méthodes pourrait être envisagée pour limiter les erreurs d'approximation.

Chapitre 6

Hachage et arbres d'ondelettes

Les travaux présentés ici constituent une méthode générale de compression pour des signaux de dimension au moins égale à deux. Le but de cette approche est de fournir une nouvelle représentation d'un signal, exprimée dans une base d'ondelettes, répondant aux limitations des méthodes actuelles. En effet, les techniques de compression basées ondelettes de l'état de l'art sont soit des méthodes de codage *Zero-Tree* soit des approches ad-hoc difficilement utilisables dans d'autres types d'application (cf. Section 3.3). Comme exposées dans nos expérimentations, les méthodes de codage *Zero-Tree* ne sont pas portables sur le matériel graphique actuel et donc, limitées à des formats de stockage uniquement. Les méthodes compatibles avec la programmation graphique souffrent de limitations sur plusieurs points. Premièrement, la taille des données n'est pas optimale car, bien souvent, le codage présenté utilise un arbre quaternaire complet, quatre fils ou aucun. Or, dans une transformée en ondelettes, l'arbre quaternaire généré est au contraire incomplet, voire irrégulier si une image par exemple contient des régions homogènes. De plus, si les données sont représentées par des réels, une quantification est obligatoire car les informations hiérarchiques et la valeur des coefficients sont représentées dans une même structure de données.

En combinant des méthodes de hachage et une transformation multi-échelle, nous proposons une structure de données qui répond à ces limitations par :

- une indépendance vis à vis du type de données ;
- une représentation d'arbres quaternaires quelconques ;
- un choix de la taille de la représentation en fonction de critères adaptatifs ;
- une structure étudiée pour le rendu sur matériel graphique.

Contrairement aux arbres d'ondelettes polynômiaux, les coefficients d'ondelettes sont ici stockés de manière atomique plutôt que par bloc. Le hachage des coefficients d'ondelettes est donc une méthode plus efficace car une transformation en paquets d'ondelettes n'est plus requise et une base d'ondelettes

plus simple peut donc être utilisée. Cette observation prend tout son sens lors du rendu où les performances sont, par conséquent, accrues mais aussi pour la compression par une sélection plus fine des coefficients.



FIGURE 6.1: Vue d'ensemble de la méthode de hachage de coefficients d'ondelettes. A partir, de données en N-Dimensions (texture, *SVBRDF*, *BTF*), le dispositif transforme et compresse les données d'origine en une représentation compacte compatible pour un rendu temps-réel sur le matériel graphique actuel.

Comme présenté sur la Figure 6.1, le dispositif est un assemblage de plusieurs méthodes. Les données utilisables sont des signaux de dimensions variables. Ici, le mode de réalisation privilégié est basé sur les *BTFs* mais cette approche n'est pas limitée à ce type de données. La première étape consiste à transformer les données dans un espace multi-échelle. Alors que les dimensions spatiales sont projetées dans l'espace ondelettes 2D, les dimensions directionnelles sont quant à elles, traitées d'une manière sensiblement différente en utilisant une transformée en ondelettes 1D pour capturer les variations anisotropes (Section 6.1). Par la suite, un nouveau type de quantification adaptative est proposé qui consiste à adapter l'espace de quantification en fonction de la distribution des données (Section 6.2). L'étape suivante constitue l'élément clé de cette méthode : les données creuses obtenues grâce aux outils précédemment appliqués sont rangés dans un ensemble de textures denses par l'utilisation de table de hachage (Section 6.3).

6.1 Transformation multi-échelle

Dans notre approche, une transformation multi-échelle est une opération de projection dans une base de fonctions qui possèdent des propriétés d'expressivité dans le domaine fréquentiel. Sur une *BTF*, nous avons envisagé dans les chapitres précédent plusieurs solutions pour définir une *BTF* multi-échelle. Cependant, ces fonctions d'apparence sont caractérisées par deux types de dimensions : les dimensions spatiales et les dimensions angulaires. Les dimensions spatiales ne soulèvent pas de problèmes particuliers pour être exprimées dans une base de fonctions multi-échelle tant le rapprochement avec les méthodes d'analyse d'images est directe. Ainsi, dans notre approche, une transformation en ondelettes 2D est appliquée suivant ces dimensions dont l'efficacité a été démontrée dans nos travaux précédents. Lorsqu'un signal directionnel doit être exprimé dans une base de fonctions sphériques, le choix de la base la plus adaptée n'est pas trivial.

Deux principales bases de fonctions sphériques multi-échelles sont communément employées dans la littérature avec les harmoniques sphériques [CMS87, WAT92] et les ondelettes sphériques [SS95, LF08]. Or, sur une *BTF*, l'échantillonnage de l'espace des directions est fixé lors du processus d'acquisition et calculé la plupart du temps de manière à ce que l'angle solide des échantillons soit constant. Les harmoniques sphériques sont un outil efficace pour projeter ce type d'échantillonnage dans une base fréquentielle. Le nombre de coefficients généré par une telle projection dépend du degré de la base dans laquelle on souhaite exprimer le signal. Sur une *BTF*, l'espace des directions comporte de nombreuses haute-fréquences de par les effets méso-structurels qui engendrent ombre et parallaxe. Ainsi, utiliser les harmoniques sphériques sur les *BTF* reviendrait à générer un nombre de coefficients de projection important car une base de degré élevé devrait être utiliser pour minimiser les erreurs de projection. Le principe est similaire pour des bases polynomiales sphériques telles que les polynômes de Zernike [KDS96, DvGNK99]. Pour représenter les haute-fréquences, les ondelettes sphériques sont plus efficaces car cette base de fonction localise spatialement les variations fréquentielles. A partir d'un domaine directionnel de base, via un maillage sphérique basse-résolution, des subdivisions géométriques sont appliquées jusqu'à ce que les données sphériques puissent être représenter avec un minimum d'erreurs. L'analyse en ondelettes géométriques sur le maillage haute-résolution ainsi construit permet de décorréler les valeurs exprimées sur chaque point du maillage dans un voisinage. En utilisant cette approche, les coefficients d'ondelettes non-nuls sont restreints aux zones où les variations, à travers les échelles, sont les plus fortes. Cependant, des informations hiérarchiques doivent aussi être stockées pour faire correspondre un coefficient à une fonction de base en particulier. Par conséquent, plus la profondeur de la base est importante plus les informations hiérarchiques sont nombreuses. Un compromis doit donc être trouvé pour minimiser à la fois les informations hiérarchiques mais aussi le nombre de coefficients.

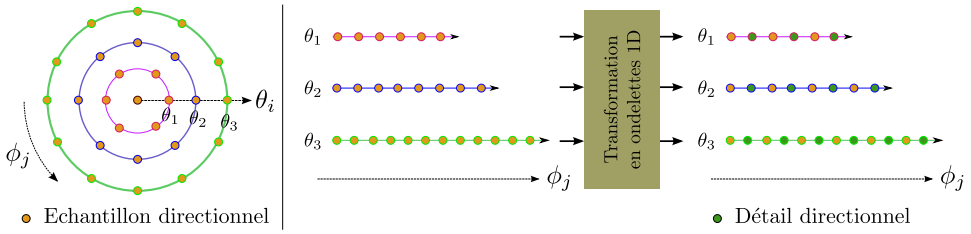


FIGURE 6.2: Transformation de l'espace des directions. Chaque échantillon directionnel est linéarisé suivant l'angle d'élévation θ. Une transformation en ondelettes 1D est ensuite appliquée suivant ϕ.

Dans cet esprit, une projection partielle de l'espace des directions est pro-

posée. Cette transformation s'applique dans l'espace des points de vue, dans l'espace des directions de la lumière ou sur l'un et l'autre de manière consécutive. Comme exposés sur la Figure 6.2, les échantillons ayant le même angle d'élévation θ sont considérés comme un signal 1D, suivant l'angle azimutal ϕ. Dans les *BTFs* disponibles publiquement, les échantillons suivant l'angle azimutal sont capturés de manière régulière. Ainsi, une transformation en ondelettes de première génération en 1D peut être appliquée pour capturer les variations anisotropes de l'espace des directions d'une BTF. Notons que cette transformation est sans perte contrairement à une projection avec les harmoniques sphériques et les ondelettes géométriques. De plus, la complexité d'une telle opération est très faible, propriété intéressante pour ne pas pénaliser la reconstruction lors du rendu.

Dans notre approche, la transformation en ondelettes dans le sens de l'anisotropie est appliquée une seule fois par espace directionnel. Le schéma sur la Figure 6.3 exprime de manière symbolique la transformation sur deux points de vue contenant plusieurs échantillons directionnels de l'espace des directions de la lumière. Dans cet exemple, la première étape consiste à transformer l'espace des directions de la lumière, notée TL. Il en résulte un ensemble d'échantillons d'approximation pour l'espace des directions de la lumière pour chaque point de vue et des détails qui encodent la différence entre original et approximation. Ces échantillons d'approximation sont ensuite décorrélés entre eux suivant les directions de points de vue, transformation notée TV, ce qui génèrent ainsi une nouvelle approximation et de nouveaux coefficients de détails communs aux deux points de vue. Ainsi, nous disposons de trois types de coefficients directionnels :

- les coefficients d'approximation de vue et de lumière,
- les coefficients de détails de vue et de lumière,
- les coefficients de détails de lumière,

où l'amplitude des coefficients décroît dans cet ordre et forme ainsi une hiérarchie dans l'espace angulaire. Chaque échantillon sur les *BTFs* est représenté par une texture sur laquelle un transformation en ondelettes 2D est appliquée, décomposition notée TS.

Le Tableau 6.1 identifie le nombre d'échantillons dans chacune des dimensions d'une BTF après transformation multi-échelle à partir de l'échantillonnage des *BTFs* de l'université de Bonn [BTFa]. Ces résultats montrent l'ensemble des configurations possibles avec notre système pour projeter les dimensions angulaires et spatiales dans un espace multi-échelle. En utilisant une méthode de seuillage des coefficients, les taux de compression sont influencés par la configuration choisie. Une décorrélation de l'ensemble des dimensions suivant la configuration $TS + TL + TV$ laisse présumer de meilleurs résultats.

6.2 Quantification adaptative

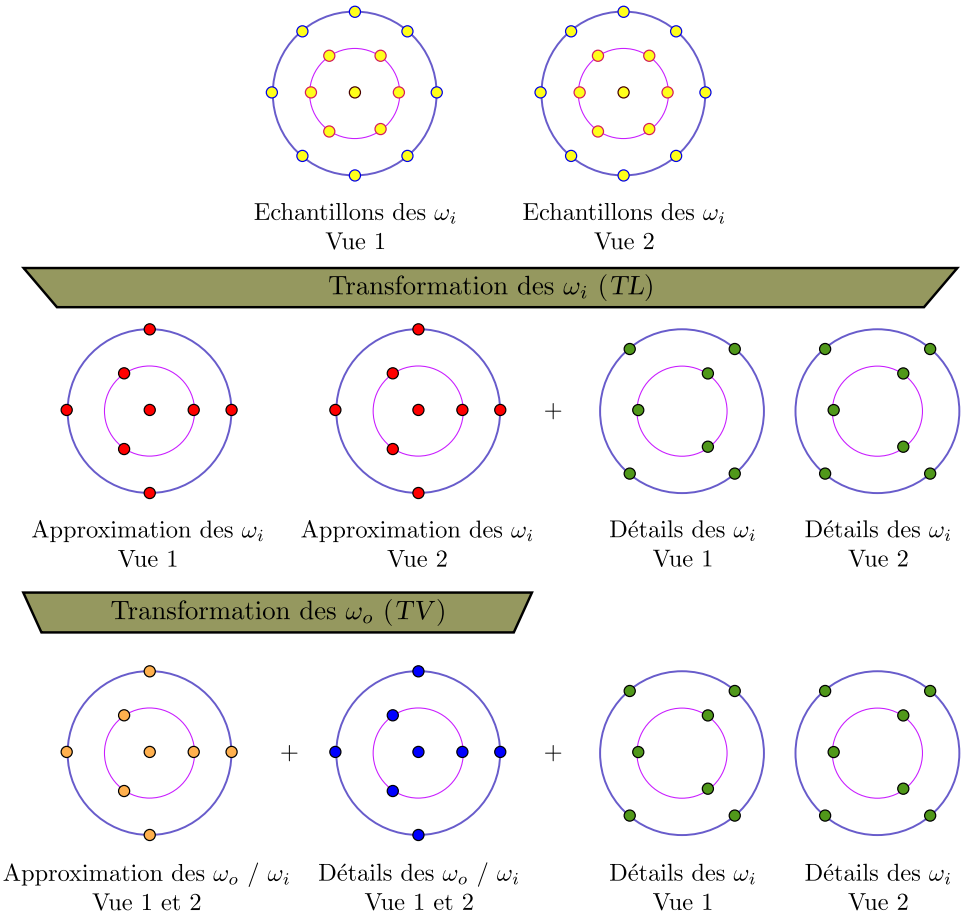


FIGURE 6.3: Transformation symbolique de l'espace des directions pour deux points de vue représentés pour différentes directions de la lumière. Une première transformation est réalisée suivant les directions de la lumière. Les coefficients d'approximation de chaque point de vue issus de cette transformation sont ensuite décomposés entre eux pour former une hiérarchie de coefficients angulaires.

Dans les méthodes de traitement du signal, la quantification est un procédé d'approximation d'un signal continu par un ensemble de valeurs discrètes de taille restreinte. Le pas de quantification détermine une segmentation du domaine de départ dans lequel une valeur discrète dans le domaine de quantification est associée à un intervalle dans le domaine de départ. Lorsque ce pas est constant dans le domaine d'origine, on parle alors de quantification uniforme. Si le pas de quantification autour de zéro est deux fois plus important que les autres intervalles du domaine départ, on qualifie cette approche de quantification avec zone morte (cf. Section 4.2). Enfin,

	Approximation $\vec{\omega}_o/\vec{\omega}_i$	Détail $\vec{\omega}_o/\vec{\omega}_i$	Détail $\vec{\omega}_i$
$TL + TV$			
η_o	41	40	81
η_i	41	41	40
η_x	256×256	256×256	256×256
Nb. Tex.	1681	1640	3240
TL			
η_o	81	-	81
η_i	41	-	40
η_x	256×256	-	256×256
Nb. Tex.	3321	-	3240
$TS + TL + TV$			
η_o	41	40	81
η_i	41	41	40
η_x Approx.	1×1	1×1	1×1
η_x Détail	$256 \times 256 - 1$	$256 \times 256 - 1$	$256 \times 256 - 1$
Nb. Tex.	1681	1640	3240
$TS + TL$			
η_o	81	-	81
η_i	41	-	40
η_x Approx.	1×1	-	1×1
η_x Détail	$256 \times 256 - 1$	-	$256 \times 256 - 1$
Nb. Tex.	3321	-	3240

TABLE 6.1: Échantillonnage des dimensions d'une *BTF* après transformation multi-échelle. Dimensions originales d'une *BTF* issue de l'université de Bonn [BTFa] : $\eta_o = 81$, $\eta_i = 81$ et $\eta_x = 256 \times 256$. TL : transformation suivant les directions de la lumière. TV : transformation suivant les directions du point de vue. TS : transformation suivant les dimensions spatiales.

lorsque la quantification est guidée par la distribution des valeurs du domaine de départ, la quantification est dite adaptative. Cette dernière approche permet de minimiser au mieux les erreurs de quantification en fonction du signal d'entrée.

Pour définir une fonction de quantification optimale, la méthode consiste à minimiser une distorsion définie par une métrique d'erreur. En définissant l'erreur quadratique moyenne comme métrique, une quantification optimale est proposée par Lloyd [Llo82]. Ce type de quantification est construit en fonction de la distribution des coefficients du domaine de départ et donc n'est pas adapté à n'importe quel type de signal. Dans la Section 4.2, nous avons démontré qu'à partir d'images de distributions différentes, la distribution de la transformée en ondelettes de ces images était sensiblement la même.

L'idée retenue ici est donc de fournir une quantification multi-échelle adaptée à la distribution de la transformée en ondelettes d'un signal. De plus, par soucis de scalabilité des données, nous définissons un critère de qualité sur un signal exprimé dans une base d'ondelettes basé sur le seuillage des coefficients d'ondelettes.

6.2.1 Critère qualitatif

Le critère de qualité, tel que développé ici, est un pourcentage de l'énergie totale du signal. Sur une BTF, l'énergie est définie comme :

$$E = \sum_{\vec{\omega}_o \in \mathbb{W}_o} \sum_{\vec{\omega}_i \in \mathbb{W}_i} \sum_{x \in \mathbb{T}} BTF\,(x, \vec{\omega}_o, \vec{\omega}_i)^2.$$

La transformation en ondelettes sur une BTF permet d'exprimer ces données de manière progressive. En triant les coefficients dans une liste par ordre croissant d'amplitude (en valeur absolue), l'opération qui consiste à supprimer un pourcent d'énergie tout en minimisant l'erreur quadratique moyenne revient à supprimer les n premiers éléments de la liste telle que la somme des carrés de ces n coefficients soit égale à $0,01 * E$. Le critère de qualité est donc mis en place comme un pourcentage de l'énergie totale du signal : une qualité de 50%, par exemple, correspond à une approximation de la BTF dont l'énergie est égale à $0,5 * E$. Cette approche revient à déterminer une valeur seuil associée à chaque pourcentage de qualité pour laquelle tout coefficient inférieur est considéré comme nul. Pour ce calcul, la méthode retenue est de construire un histogramme des coefficients pour représenter la contribution énergétique de chaque classe de l'histogramme. En construisant les classes de manière suffisamment fine dans une échelle logarithmique, les valeurs seuils sont calculées en parcourant l'histogramme dans un ordre croissant et, à chaque intervalle de 1% de l'énergie, la valeur courante de la classe est considérée comme une valeur seuil. L'avantage de cette méthode de calcul est que, à tout moment, seul un élément de la BTF (i.e. une texture) est chargé en mémoire pour les calculs d'histogramme alors que l'algorithme naïf nécessiterait de charger l'ensemble des coefficients en mémoire. Cette approche est donc une méthode de calcul efficace pour l'ajustement du critère de qualité.

La Figure 6.4 illustre le nombre de coefficients présent dans la représentation, un pourcentage du nombre totale de coefficients, par rapport au pourcentage de qualité défini. Le calcul est effectué ici en considérant la représentation au niveau de qualité 0 comme l'ensemble des coefficients d'approximation de la BTF (cf. tableau 6.1). Ainsi, suivant le type de décomposition utilisée, l'ensemble minimal de coefficients est plus ou moins important. On remarque que la taille d'une représentation pour un pourcentage de qualité donné varie selon le type de transformation appliqué. Les résultats montrent que la quantité de coefficients de faible amplitude est maximale si

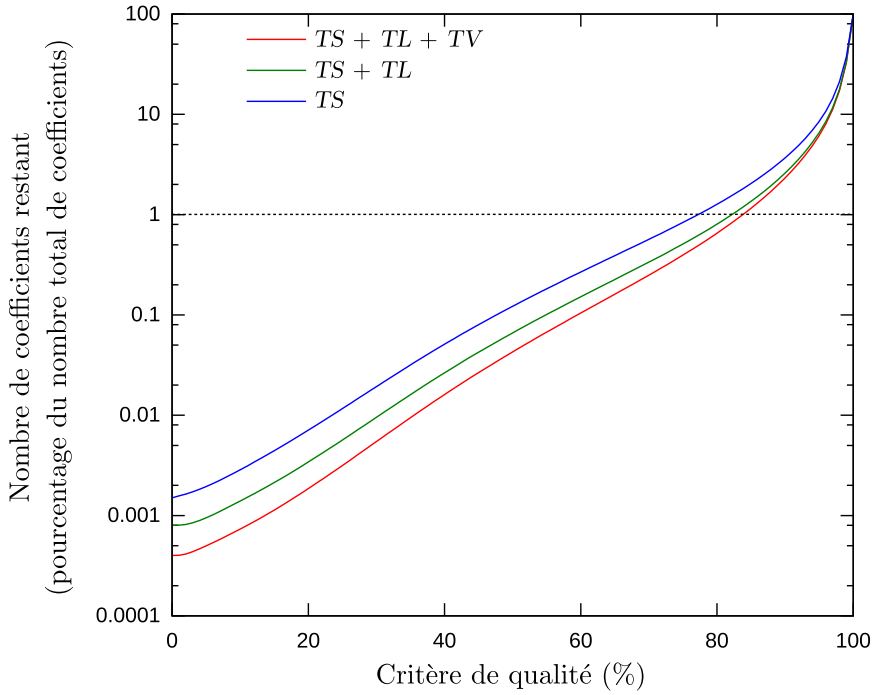


FIGURE 6.4: Critère de qualité défini sur une *BTF*. Le pourcentage de qualité est représenté en abscisse et le nombre de coefficients restant dans la représentation en ordonnée. Différentes décompositions en ondelettes sont proposées pour juger de l'efficacité de chaque transformation.

l'on applique une transformation sur l'ensemble des dimensions de la *BTF*. Cette observation valide donc aussi l'approche choisie pour la transformation de l'espace des directions qui capture la redondance inter-directionnelle. De plus, ces résultats montrent qu'en gardant seulement 1% des coefficients, l'énergie totale de la *BTF* est, dans le meilleur des cas, de 85%. La fonction de qualité révèle donc des propriétés intéressantes pour la compression où un seuillage relativement faible entraîne une forte réduction du nombre de coefficients.

6.2.2 Quantification

Les *BTFs* sont généralement exprimées dans un espace de couleur où chaque composante est un réel sur 32 bits pour des facilités de calcul, voire pour des représentations en haute dynamique. En supplément au seuillage de coefficients tel que présenté précédemment, la représentation d'une *BTF* par un nombre plus restreint de bits par composante est souvent nécessaire pour obtenir de forts taux de compression. L'idée principale retenue dans notre schéma de quantification consiste à définir une fonction de déformation des

coefficients pour chaque niveau de décomposition qui réalise une conversion non-uniforme des valeurs réelles sur 32 bits vers une valeur quantifiée entière sur 8 bits. La fonction de déformation est guidée par la distribution cumulative des coefficients d'ondelettes pondérée par l'amplitude des coefficients telle que présenté sur la Figure 6.5.

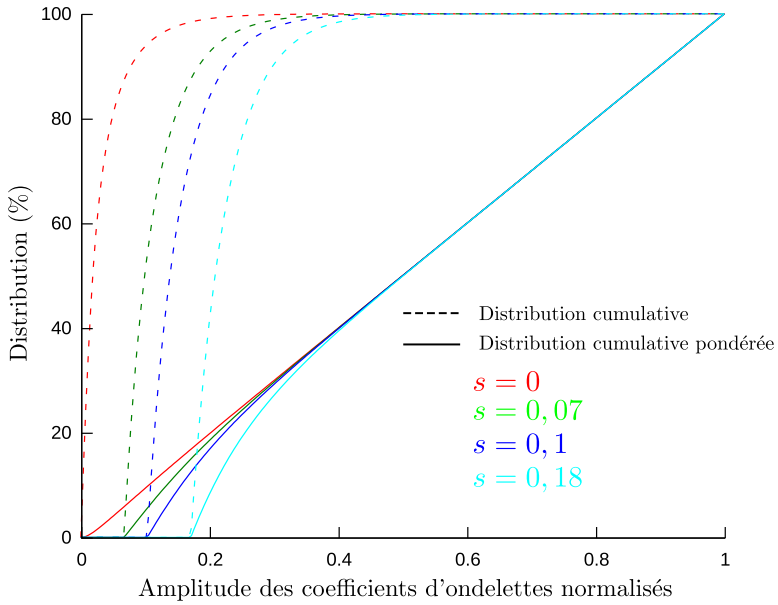


FIGURE 6.5: Distribution cumulative et distribution cumulative pondérée des coefficients de détail au niveau de décomposition 1 d'une *BTF* après transformation en ondelettes (TS) suivant plusieurs valeurs seuils s.

Cette approche permet de bien estimer l'importance relative de chaque coefficient et ainsi de réduire la quantité d'erreur liée à la quantification. Comme présentée sur la Figure 6.5, la distribution cumulative des coefficients d'ondelettes permet de juger de la prépondérance des valeurs de faible amplitude dans une décomposition en ondelettes. Cependant, le nombre de coefficients dans une classe d'amplitude n'est pas le seul critère à prendre en compte pour minimiser les erreurs de quantification. En effet, l'amplitude des coefficients est aussi un facteur d'importance minimisation : une erreur relative de 1% sur un coefficient d'amplitude 1000 est aussi importante que la même erreur sur 1000 coefficients d'amplitude 1. De cette observation, un estimateur logique est de pondérer la distribution cumulative des coefficients par l'amplitude notée *distribution cumulative pondérée* sur la Figure 6.5.

En analysant cet estimateur pour un grand nombre de données, les résultats ont montré que l'allure des courbes de distribution cumulative pondérée est globalement la même. La fonction est fortement non-uniforme pour les coefficients de faible amplitude mais devient pratiquement uniforme pour ceux

de forte amplitude. Nous proposons donc une approximation de ces courbes par une fonction polynomiale par morceaux contrôlée par deux paramètres. Le premier, p, influe sur la zone morte de l'espace de quantification. Le deuxième, q, correspond au point de jonction entre la partie non-uniforme et uniforme de la courbe. Ainsi, la fonction de quantification Q est définie comme :

$$Q(x) = (x < p)0 : (x < q)?\frac{q^2 * (x - p)}{q^2 - 2pq + px} : x, \quad x \in [0 : 1].$$

Les paramètres p et q dépendent du critère de qualité expliqué précédemment. Pour chaque pourcentage de qualité, une seuil s est déterminé comme la valeur minimale d'un coefficient d'ondelettes. Ainsi, les paramètres sont calculés tels que :

- $Q(s) = \Delta$, avec Δ, le pas de quantification, qui permet d'initialiser p ;
- q est fixé empiriquement à la valeur suivant l'axe des abscisses où la distribution cumulative est égale à $0,99$.

Les résultats d'approximation sont présentés sur la Figure 6.6. Ces courbes montre que l'approximation polynomiale par morceaux et l'initialisation des paramètres p et q permettent d'approximer quasi-parfaitement la distribution cumulative pondérée des coefficients d'ondelettes suivant différents critères de qualité.

La fonction de déformation a été choisie de manière à être inversible pour permettre la quantification inverse. Sa formulation analytique, pour les mêmes paramètres p et q, est :

$$Q^{-1}(x) = (x < q)?\frac{p * q * (q - x) + (q - p) * q * x}{q * q - p * x} : x.$$

La fonction de quantification inverse est représentée sur les exemples de la Figure 6.6.

La fonction de quantification est appliquée sur les coefficients d'ondelettes normalisés. L'espace de quantification engendré est par la suite subdivisé de manière uniforme suivant un pas de quantification Δ dépendant du nombre de bits par composante. Dans notre approche, chaque coefficient d'ondelettes est quantifié sur 8 bits : 1 bit de signe et 7 pour la quantification. Dans cette configuration, l'erreur relative globale est inférieure à 1% sur la plupart des matériaux pour une différence visuelle nulle.

6.3 Hachage des coefficients

6.3.1 Présentation

En informatique, une *table de hachage* est une structure de données qui utilise une *fonction de hachage* pour stocker de manière plus efficace un

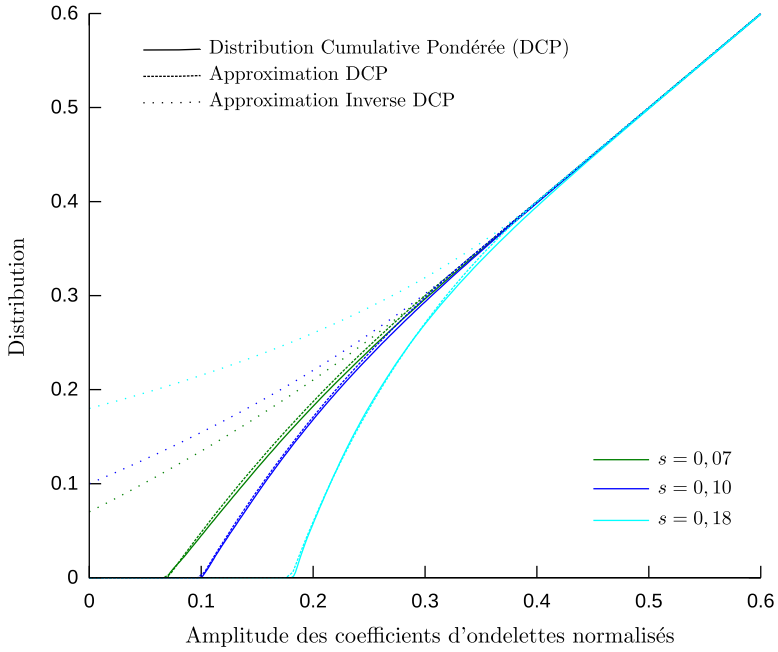


FIGURE 6.6: Distribution cumulative pondérée des coefficients de détail au niveau de décomposition 1 d'une *BTF* après transformation en ondelettes (TS) suivant plusieurs valeurs seuils s, l'approximation de la DCP et sa fonction inverse. *(Zoom sur la partie non-uniforme)*

ensemble de couples { clé, valeur } d'un domaine particulier. La fonction de hachage permet de transformer une clé en une adresse dans la table de hachage pour y stocker la valeur correspondante. Dans le cas idéal, chaque clé du domaine de départ possède une adresse unique, par la fonction de hachage, dans la table de hachage ; on parle alors de hachage parfait. Lorsque les images de plusieurs clés par la fonction de hachage sont identiques, il y a alors *collision* entre les différents éléments qu'il faut résoudre.

Dans la littérature, on retrouve de nombreuse méthodes pour résoudre les collisions. La méthode la plus utilisée consiste à considérer chaque élément de la table de hachage comme une liste chaînée. Ainsi, à chaque collision, l'élément perturbateur est ajouté comme un nouvel élément de la liste à l'emplacement désigné par la fonction de hachage. La complexité d'un tel algorithme est linéaire, $\mathcal{O}(n)$, et pour des raisons de structure de données, est difficilement utilisable directement sur les processeurs graphiques. En informatique graphique, des travaux ont été proposés pour permettre un hachage parfait sur des images ou une grille 3D creuses suivant deux techniques de hachage : le hachage par décalage [LH06] et le hachage multi-adresse [ASA+09]. La première méthode consiste à utiliser une valeur, un décalage partagé entre plusieurs éléments, en plus de la fonction de hachage pour

gérer les collisions. La deuxième technique utilise n fonctions de hachage pour envisager le stockage d'un élément à n adresses différentes et ainsi, éviter les collisions par un choix multiple d'emplacement. La recherche d'un élément dans la table de hachage est réalisée en temps constant $\mathcal{O}(1)$ par ces deux méthodes et les structures de données associées, tableaux de une, deux ou trois dimensions, sont utilisables sur le matériel graphique. Le choix de la méthode la plus adaptée dépend du type d'utilisation des données. La construction de la table de hachage peut être réalisée en temps interactif [ASA+09] avec la méthode du multi-adressage alors que la méthode par décalage nécessite un temps relativement long pour tendre vers un hachage parfait. Cependant, les taux de remplissage (i.e. la densité de la table de hachage) sont meilleurs avec la méthode de décalage, supérieurs à 90%, que par le hachage multi-adresse, de l'ordre de 70%.

Dans notre approche, les éléments à stocker sont un ensemble de textures bi-directionnelles dont la densité originale ne permet pas d'utiliser les méthodes de hachage de manière efficace. Cependant, après projection dans une base d'ondelettes et application d'un critère de qualité plus ou moins destructif, les textures sont des éléments fortement creux pour lesquelles les techniques de hachage sont très efficaces. Ici, les taux de compression sont plus importants que le temps de construction dans la mesure où le codage est effectué hors-ligne. La méthode par décalage semble donc un meilleur choix pour la construction de tables de hachage sur les textures de la BTF.

La stratégie choisie est donc de stocker un ensemble de { clé, valeur } où la clé correspond à une coordonnée de texture et la valeur au coefficient d'ondelettes du pixel à ces coordonnées. Une coordonnée de texture est dans ce cas un identifiant unique et, sur une BTF, nous choisissons de définir une telle coordonnée comme un vecteur en trois dimensions où les deux premières composantes sont les coordonnées spatiales d'un pixel et la troisième composante, un identifiant du point de vue et de la direction de la lumière associé à la texture. Seuls les coefficients d'ondelettes non-nuls sont stockés dans la table de hachage. Les coordonnées de texture associées à chaque coefficient non-nul représentent un sous-ensemble de l'ensemble des coordonnées de texture d'une BTF. Ce sous-ensemble doit être renseigné explicitement par une *carte du domaine* identifiant si un coefficient est stocké ou non dans la table de hachage. L'algorithme se décompose donc en deux étapes où la première consiste à construire la carte du domaine des coefficients non-nuls et la deuxième permet de stocker les coefficients identifiés comme présents dans la carte du domaine dans la table de hachage.

6.3.2 Carte du domaine

La méthode de hachage permet de stocker les coefficients d'ondelettes de l'ensemble des textures, un sous-ensemble de pixels, d'une manière quasi-optimale. Cependant, lors de la reconstruction, il est nécessaire d'identifier

quels sont les coefficients (i.e. coordonnées de texture) qui appartiennent au sous-ensemble de coefficients d'ondelettes non-nuls. L'approche retenue ici consiste à coder un arbre de décomposition en ondelettes comme un arbre de bits où un noeud de l'arbre dont le bit est égal à 0 (resp. 1) indique que le coefficient à ce niveau de hiérarchie est exclu (resp. inclus) dans la représentation.

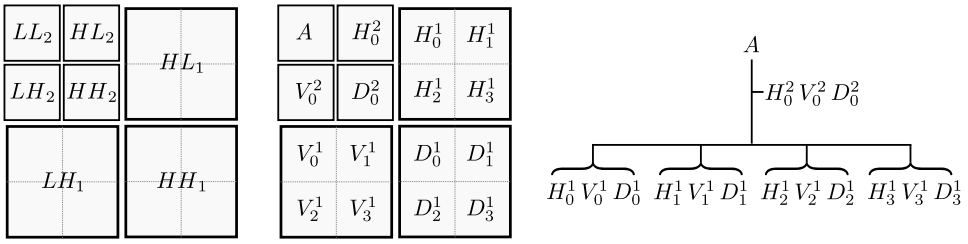


FIGURE 6.7: Décomposition en ondelettes d'une image 4×4 après deux niveaux de décomposition. A gauche, les sous-bandes spatio-fréquentielles générées par la décomposition. Au milieu, à chaque pixel des sous-bandes est associé un identifiant H_j^i, V_j^i ou D_j^i correspondant respectivement aux coefficients de détail horizontaux, verticaux et diagonaux ; l'identifiant A correspond au coefficient d'approximation. A droite, l'arbre de décomposition en ondelettes des coefficients dans la configuration que nous avons choisie.

Le coût du codage de chaque arbre de décomposition en ondelettes pour chaque texture composant une BTF est de $\eta_o \times \eta_i \times \eta_x$ bits, soit, sur une BTF de l'université de Bonn [BTFa], 51 Mo. Cette approche est donc d'une complexité en mémoire trop grande pour le codage de la hiérarchie uniquement. Pour améliorer les résultats quantitatifs, nous proposons une factorisation des éléments de l'arbre de décomposition en regroupant dans chaque noeud de l'arbre les éléments spatialement dépendants. Comme présentés sur la Figure 6.7, les détails horizontaux, verticaux et diagonaux qui participent à la reconstruction d'une même région spatiale sont regroupés dans le même noeud. En regroupant maintenant les informations hiérarchiques par branche (i.e. par groupe de quatre noeuds de trois coefficients chacuns), une branche de l'arbre est codée sur 12 bits et décrit sa *configuration*, 0 si un coefficient est nul, 1 sinon pour chaque bit. Cependant, il est possible de factoriser les configurations par vraisemblance et ainsi, en autorisant une famille de 2^8 configurations parmi les 2^{12} possibles, une branche peut ainsi être codée sur 8 bits. Le coût du codage de la hiérarchie complète est ainsi de 34 Mo, un compromis entre complexité de codage et taille de la hiérarchie, plus l'ajout d'une table de correspondances dont la taille est négligeable.

La factorisation des configurations d'une branche est réalisée post-seuillage lors du choix de la qualité de représentation. Pour chacune des 2^{12} possibilités, un histogramme est construit à partir de la BTF transformée pour

référencer le nombre d'occurences de chaque configuration. En triant les configurations par ordre décroissant de leur valeur dans l'histogramme, il est possible d'identifier les n configurations présentes dans la représentation. Empiriquement, nous avons choisi de fixer le nombre de configurations possibles à 2^8. Si $n < 2^8$ alors, le codage de la hiérarchie peut être opéré en associant chaque configuration à une clé comprise entre 0 et n. Si $n > 2^8$, les $n - 2^8$ configurations supplémentaires doivent être prises en compte. Pour se faire, une stratégie conservatrice est utilisée en trouvant pour chaque configuration à exclure une configuration compatible par ajout de coefficients nuls. Les coefficients ainsi ajoutés sont qualifiés de *faux-positifs* car ils sont présents dans la représentation mais considérés comme nuls. Dans notre approche, l'ajout de faux-positifs n'est pas contraignant si la quantité de coefficients ajoutés reste faible. Or, si l'on effectue la factorisation de configurations sur l'ensemble de la BTF, le nombre de faux-positifs est important, environ 10%. Pour en limiter le nombre, la factorisation est réalisée localement dans un voisinage angulaire et spatial.

Ainsi, l'algorithme de factorisation est un algorithme glouton réalisé par point de vue. La factorisation est, à l'initialisation, accomplie pour chaque point de vue sur l'ensemble du domaine spatiale. Travailler par point de vue permet de s'abstraire des effets de parallaxe et donc de privilégier la cohérence spatiale des configurations. Si après cette première itération, le nombre de faux-positifs est trop important, on subdivise les régions spatiales par un facteur 4 et la factorisation est recalculée sur chaque point et chaque région spatiale indépendemment. Tous les coefficients basse-fréquences qui ne peuvent être associés à une région spatiale, car plus grossiers, sont considérés comme appartenant à la représentation. Regrouper la factorisation par région spatiale permet d'accroître la cohérence entre les configurations de chaque branche de par l'aspect spatio-fréquentiel d'un coefficient d'ondelettes. Dans notre approche, nous avons limité le nombre de faux-positifs à 3%. Ce résultat est généralement obtenu pour un nombre de régions spatiales égal à 16.

6.3.3 Hachage des coefficients

Le hachage des coefficients d'ondelettes tel que utilisé ici est proche de la méthode de hachage présentée par Lefebvre *et al.* [LH06]. Cependant, cette dernière est limitée au stockage d'images ou de voxels creux. Or, ici, grâce à la transformation en ondelettes, le domaine applicatif d'une telle méthode est généralisé à l'ensemble des images, ou autres, car les coefficients dans l'espace ondelettes sont creux. De plus, nous présentons une variante de l'algorithme originel pour nous permettre de modeler la taille de la table de décalage de manière plus fine et ainsi améliorer les résultats quantitatifs. De plus, la méthode de Lefebvre *et al.* fonctionne pour une densité de coefficients non-nuls dans l'image de départ relativement faible. L'optimisation que nous apportons permet de stocker des images de densité plus élevée.

Le principe général est illustré sur la Figure 6.8 sur une image. Deux fonctions de hachage indépendantes sont utilisées pour faire correspondre un pixel de coordonnées $\mathbf{p_i} = (u, v)$ dans une image $\mathcal{I}$ de taille $m \times m$ à une adresse, d'une part, $\mathbf{p_d}$ dans la table de décalage $\mathcal{D}$ avec $h_2(\mathbf{p_i}) = \mathbf{p_d}$ et d'autre part, $\mathbf{p_h}$ dans la table de hachage $\mathcal{H}$ en utilisant $h_1(\mathbf{p_i}) = \mathbf{p_h}$. Les fonctions de hachage utilisées sont des fonctions quadratiques du type :

$$h(\mathbf{p_i}) = \begin{cases} \alpha * u^2 + \beta * u + \gamma \\ \alpha * v^2 + \beta * v + \gamma \end{cases}$$

où α, β et γ sont trois entiers (< 256) choisis aléatoirement. Le résultat de ces fonctions est utilisé modulo la taille de la table, n pour la table de hachage et k pour la table de décalage.

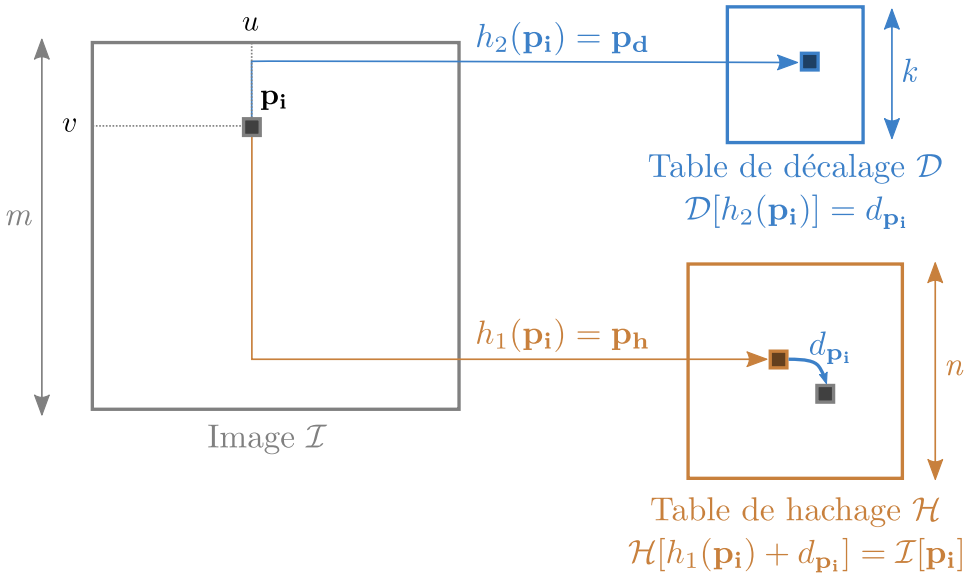


FIGURE 6.8: Méthode de hachage par décalage sur une image. La valeur d'un pixel non-nul $\mathbf{p_i} = (u, v)$ de l'image $\mathcal{I}$ est retrouvée dans la table de hachage $\mathcal{H}$ à l'adresse $h_1(\mathbf{p_i}) + d_{\mathbf{p_i}}$, où $d_{\mathbf{p_i}}$ est une valeur de décalage pour gérer les collisions stockée dans la table de décalage $\mathcal{D}$ à l'adresse $h_2(\mathbf{p_i})$ avec h_1 et h_2 deux fonctions de hachage indépendantes.

Les fonctions h_i sont surjectives c'est à dire que un ou plusieurs pixels de l'image $\mathcal{I}$ ont les mêmes coordonnées dans la table de décalage ou dans la table de hachage par h_1 ou h_2. Par conséquent, sauf cas exceptionnel où h_1 est injective, l'utilisation de la table de hachage uniquement n'est pas suffisante pour gérer les collisions. La table de décalage est donc une solution pour résoudre les imperfections de la fonction h_1. Ainsi, tous les pixels de l'image $\mathcal{I}$ qui appartiennent à l'ensemble formé par $h_2^{-1}(\mathbf{p_d})$, pour un pixel

$\mathbf{p_d}$ donné dans $\mathcal{D}$, partagent la même valeur de décalage pour solutionner les collisions.

L'algorithme permettant un tel hachage des données est un algorithme itératif glouton où à chaque tour de boucle, la taille de la table de décalage change. Le principe consiste à chercher pour tous les pixels non-nuls de l'image, un décalage, partagé entre plusieurs de ces pixels, tel qu'il n'y ait pas de collisions dans la table de hachage. La taille de $\mathcal{H}$, $n \times n$, est initialisée tel que le nombre de pixels dans $\mathcal{H}$ soit égal aux nombres de pixels non-nuls dans $\mathcal{I}$, par excès. La taille de $\mathcal{D}$, $k \times k$, est initialisée tel que le nombre de pixels de $\mathcal{D}$ soit égal à 5% du nombre de pixels non-nuls dans $\mathcal{I}$. L'algorithme se déroule comme suit :

1. Initialiser $\mathcal{H}[\mathbf{p_h}] = 0$, $\forall \mathbf{p_h} \in \mathcal{H}$.
2. Calculer la carte $h_2^{-1}(\mathbf{p_d})$, $\forall \mathbf{p_d} \in \mathcal{D}$;
3. $\forall \mathbf{p_d} \in \mathcal{D}$:
   (a) Trouver une valeur de décalage $d_{\mathbf{p_i}}$ tel que
   $$\mathcal{H}[h_1(\mathbf{p_{i_j}}) + d_{\mathbf{p_i}}] = 0,\ \forall j \mid \mathbf{p_{i_j}} \in h_2^{-1}(\mathbf{p_d}).$$
   (b) Si $d_{\mathbf{p_i}}$ existe alors
      i. $\mathcal{H}[h_1(\mathbf{p_{i_j}}) + d_{\mathbf{p_i}}] = \mathcal{I}(\mathbf{p_{i_j}}),\ \forall j$
      ii. retourner en 3 ;
   (c) Sinon,
      i. agrandir k,
      ii. retourner en 1.

Ce procédé itératif converge vers une solution quand la table de décalage est suffisamment grande pour que tous les pixels non-nuls de l'image $\mathcal{I}$ puissent être associés à une adresse unique dans la table de hachage. L'algorithme appliqué ainsi n'a que très peu de chance de converger vers une solution où k est quasi-optimal. Une solution plus efficace est de trier la carte $h_2^{-1}(\mathbf{p_d})$ par ordre décroissant d'occurrence. Ainsi, le plus grand groupe de pixels de l'image $\mathcal{I}$ partageant le même décalage en $\mathbf{p_d}$ est assigné en premier dans la table de hachage ce qui laisse les plus fortes probabilités de collision sur des groupes de petite taille plus faciles à stocker.

Pour trouver la valeur de décalage des pixels de $\mathcal{I}$ appartenant à $h_2^{-1}(\mathbf{p_d})$ pour un $\mathbf{p_d} \in \mathcal{D}$ donné, la solution retenue est de faire un tirage aléatoire de la valeur de décalage pour un nombre d'essais limité. Si les essais sont infructueux, l'ensemble des valeurs de décalage non essayées est testé jusqu'à trouver une solution. Si aucune solution n'est trouvée, alors il y a collision et la taille k de la table de décalage est agrandie d'un facteur corrélé au nombre de pixels traité par rapport au nombre de pixels total.

Afin de multiplier les chances de trouver une solution pour un k relativement petit, nous introduisons une méthode de multi-adressage des valeurs.

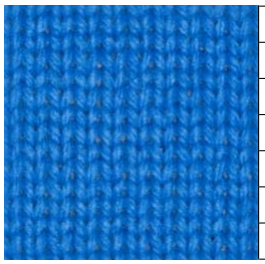
En effet, lorsqu'une valeur de décalage est affectée pour stocker un ensemble de pixels dans la table de hachage, l'image des coordonnées par h_1 des pixels non-nuls appartenant à $h_2^{-1}(\mathbf{p_d})$ restent identique à une valeur de décalage près. Un plus grand nombre de hachages est possible en utilisant une famille de fonctions de hachage plutôt qu'une seule fonction h_1 pour d'une part, résoudre les collisions plus efficacement et d'autre part, minimiser la taille de la table de décalage. Ainsi, notre algorithme permet, lors du calcul de la valeur de décalage, de choisir parmi les 2^n fonctions de hachage pour stocker un ensemble de pixels dans $\mathcal{H}$. De plus, cette approche permet de fixer au préalable une taille k de $\mathcal{D}$ et d'assigner les valeurs dans la table de hachage soit par des techniques de re-hachage (i.e. redéfinir une famille de fonctions de hachage) soit en augmentant le nombre de fonctions de hachage.

6.4 Résultats

Pour effectuer le hachage des coefficients d'ondelettes, les coefficients sont organisés par groupe de vraissemblance et la disposition dépend du type de décomposition appliqué. Cette structuration doit être la plus homogène possible pour permettre un stockage en mémoire vidéo dans un nombre restreint de textures. Si seule une transformation spatiale des données est effectuée, alors, les textures peuvent être regroupées par point de vue en disposant chacune des textures appartenant au même point de vue dans une texture carrée de taille $\left\lceil \sqrt{\eta_i \times \eta_x} \right\rceil$. Dans le cas de l'échantillonnage des données de l'université de Bonn, la taille d'une texture par point de vue est 9×256, soit 9×9 textures. Chaque point de vue est ainsi uniforme sur le nombre de coefficients : la transformation en ondelettes est la même pour tous les points de vue. Le procédé est similaire si une transformation suivant l'espace des directions de la lumière est appliquée. Cependant, lors d'une transformation suivant les points de vue, le nombre potentiel de coefficients entre chaque groupe n'est plus équilibré : certains points de vue sont des points de vue d'approximation avec, en théorie, une densité de coefficients supérieure aux points de vue de détail. Dans ce contexte, les textures sont regroupées par point de vue et par type de coefficients et encodées suivant trois structures de données différentes (deux uniquement si la transformation suivant l'espace lumière n'est pas appliquée).

Ainsi, les résultats que nous présentons dans un premier temps portent sur l'optimisation apportée dans la construction de la table de hachage. Le tableau 6.2 illustre les performances de l'algorithme de hachage en fonction du nombre de fonctions de hachage choisi pour assigner les données. La taille de la table de décalage k diminue à mesure que le nombre de fonctions utilisées augmente. Dans cet exemple, nous n'avons pas réussi à faire converger l'algorithme en utilisant une seule fonction tout en maintenant une taille fixe sur la table de hachage ; La densité des coefficients non-nuls dans le domaine

de départ étant trop importante. Le multi-adressage permet donc d'étendre les possibilités de cette technique à des représentations moins creuses.

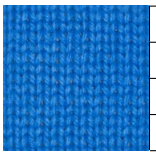


Nb. fonctions de hachage	k
1	∞
2	427
4	415
8	397
16	385
32	365

TABLE 6.2: Evolution de la taille k de la table de décalage en fonction du nombre de fonctions de hachage choisi. Cet exemple est réalisé sur le matériau illustré [BTFa] pour un seul point de vue et différentes directions de la lumière. Seule une transformation en ondelettes suivant l'espace texture est appliquée et le critère de qualité est fixé à 95%. La taille de la table de hachage est constante : 967×967. La taille de la table de décalage diminue à mesure que le nombre de fonctions de hachage augmente.

La disposition des coefficients telle que présentée précédemment permet d'uniformiser la taille des textures générées par l'algorithme. En effet, le tableau 6.3 présente les résultats obtenus avec notre algorithme de hachage sous contrainte. Ici, seule une transformation spatiale en ondelettes est effectuée et les textures sont regroupées par point de vue. L'algorithme se décompose en deux passes. Une première consiste à appliquer le hachage sur chaque point de vue indépendemment. Cette étape permet de connaître la taille de la table de hachage et celle de la table de décalage dans une approche quasi-optimale individuellement sur chaque point de vue. La deuxième passe consiste à exécuter de nouveau l'algorithme mais en définissant une taille constante des tables pour tous les points du vue. Pour la table de hachage, cette taille est fixée à la taille maximale des tables de hachage générés lors de la première étape. Pour la taille de la table de décalage, nous choisissons la médiane des dimensions calculées lors de la première phase. Dans cette exécution sous contrainte, les collisions sont fréquentes. Dans ce cas, la taille des tables étant fixe, nous considérons que nous n'avons pas été chanceux dans le tirage aléatoire des fonctions de hachage. Ainsi, les coefficients de ces fonctions sont initialisés à nouveau avec de nouvelles valeurs aléatoires. Dans le cas général, l'utilisation de quatre fonctions de hachage par point de vue est suffisante. Les coefficients d'ondelettes sont au préalable quantifiés et la carte du domaine renseignée pour stocker l'ensemble des coefficients jugés significatifs chacun sur 8 bits. Ici, les données encodées correspondent à la luminance de la *BTF* uniquement. La table de décalage est composée de deux coefficients sur 2×8 bits par pixel pour représenter les translations

des coordonnées dans la table de hachage. Ce même algorithme est appliqué sur les résultats présentés sur les tableaux 6.4 et 6.5



Qualité (%)	$\mathcal{D}$	$\mathcal{H}$	Taille (Mo)
99	$701 \times 701 \times 81$	$1621 \times 1621 \times 81$	278
95	$455 \times 455 \times 81$	$1111 \times 1111 \times 81$	85
90	$347 \times 347 \times 81$	$827 \times 827 \times 81$	71

TABLE 6.3: Résultat de l'algorithme de hachage des coefficients d'ondelettes sur une *BTF* suivant plusieurs critères de qualité. Sur cet exemple, seule une transformation spatiale en ondelettes est appliquée.

Les résultats présentés sur le tableau 6.3 montrent les performances de l'algorithme de hachage sous contrainte sur des données où seule une transformation en ondelettes sur les dimensions spatiales est appliquée. Les résultats quantitatifs pour le stockage des coefficients sont encourageants lorsque le pourcentage de qualité est légèrement destructif. Le PSNR moyen sur les données à 99% de qualité est de 48 dB alors qu'il n'est que de 28 dB sur les données à 90%. Dans cette configuration, le ratio entre la taille des données et la qualité de la représentation n'est pas encore suffisant pour permettre un rendu de qualité sur le matériel graphique.



Qualité (%)	$\mathcal{D}$	$\mathcal{H}$	Taille (Mo)
99	$503 \times 503 \times 81$	$1180 \times 1180 \times 81$	146
95	$253 \times 253 \times 81$	$623 \times 623 \times 81$	40
90	$151 \times 151 \times 81$	$389 \times 389 \times 81$	15

TABLE 6.4: Résultat de l'algorithme de hachage des coefficients d'ondelettes sur une *BTF* suivant plusieurs critères de qualité. Une transformation spatiale en ondelettes plus une transformation de l'espace des directions de la lumière sont appliquées.

Les résultats proposés dans le tableau 6.4 sont nettement meilleurs que ceux exposés précédemment. Cette amélioration est due à l'utilisation d'une transformation en ondelettes suivant l'espace des directions de la lumière en plus de celle effectuée spatialement. Cette approche montre que l'espace des directions de la lumière est fortement corrélé et la transformation suivant cet espace est donc indispensable. Le PSNR moyen est de 48 dB pour les données à 99% et se dégrade linéairement pour atteindre 29 dB lorsque le critère de qualité est fixé à 90%.

Le tableau 6.5 indique les rÃ©sultats obtenus en utilisant utilisant une transformation complète de l'espace des directions. En conservant la même configuration, la taille de données pour un pourcentage de qualité est sensiblement identique aux résultats présentés dans le tableau 6.3. Cette similarité



Qualité (%)	Type	$\mathcal{D}$	$\mathcal{H}$	Taille (Mo)
99	A	$361 \times 361 \times 41$	$857 \times 857 \times 41$	158
	D_1	$359 \times 359 \times 40$	$845 \times 845 \times 40$	
	D_2	$371 \times 371 \times 81$	$889 \times 889 \times 81$	
95	A	$187 \times 187 \times 41$	$455 \times 455 \times 41$	45
	D_1	$183 \times 183 \times 40$	$445 \times 445 \times 40$	
	D_2	$199 \times 199 \times 81$	$485 \times 485 \times 81$	
90	A	$119 \times 119 \times 41$	$295 \times 295 \times 41$	19
	D_1	$115 \times 115 \times 40$	$289 \times 289 \times 40$	
	D_2	$125 \times 125 \times 81$	$317 \times 317 \times 81$	

TABLE 6.5: Résultat de l'algorithme de hachage des coefficients d'ondelettes sur une *BTF* suivant plusieurs critères de qualité. La transformation en ondelettes est appliquée suivant toutes les dimensions de la *BTF*. Le type de coefficient est précisé où A, représente l'approximation, D_1 sont les détails vue/lumière et D_2 les détails de la lumière uniquement (cf. tableau 6.1). La taille représente la quantité de données suivant tous les types de coefficients.

montre que l'espace des directions du point de vue n'est que très peu corrélé, spécialement sur ce matériau où les effets de parallaxe sont très importants. Cependant, décomposer la *BTF* en plusieurs sous-ensembles permet de faire converger l'algorithme de hachage beaucoup plus rapidement que dans les configurations précédentes.

Les temps d'exécution de l'algorithme sont de l'ordre de quelques minutes pour une exécution sans contrainte et jusqu'à plusieurs heures de calculs en ajoutant les contraintes sur la taille des tables. Plusieurs approches peuvent être envisagées pour améliorer les temps d'exécution. L'algorithme peut être réalisé en 3D pour éviter une exécution sous contrainte et stocker ainsi tous les coefficients dans un seul bloc de données. Bien qu'efficace pour un stockage optimale à l'aide d'une seule structure de données, cette idée est coûteuse car elle nécessite de trouver les vecteurs de décalage en 3D et de plus, oblige à charger en mémoire à la fois les données d'origine mais aussi les données encodées. Dans nos espérimentations, une approche multi-échelle est plus efficace pour séparer les données en sous-ensembles de faible taille et ainsi, générer une table de hachage par échelle de détail.

Le rendu à l'aide de ces données n'a pas été testé dans son ensemble. Toute les structures de données étant compatibles avec le matériel graphique, la portabilité ne fait aucun doute. Nous avons expérimenté le rendu pour un point de vue fixe avec transformation en ondelettes spatiales et suivant l'espace des directions de la lumière. La reconstruction via les *shaders* est de l'ordre de 80 images par seconde, performance sensiblement identique à la

reconstruction sans table de hachage. Ces travaux sont donc encourageants pour un rendu en temps réel de *BTFs* de qualité.

6.5 Bilan et approches futures

Les travaux présentés ici constituent une étude sur la définition d'un nouveau standard de compression de signaux en ondelettes permettant un rendu efficace sur matériel graphique. Actuellement, il n'existe pas de méthodes suffisamment générales et efficaces pour permettre à la fois, des taux de compression élevés mais aussi, une visualisation performante sur matériel graphique pour ce type de signaux. Des éléments de réponse sont donc avancés ici où chaque étape du processus est présentée sur un mode de fonctionnement privilégié mais dont la généralisation est implicite. De nombreuses autres configurations sont à envisager pour définir une méthode générale : nombre de bits par pixel lors de la quantification, variétés des données à traiter ou une comparaison exhaustive avec les méthodes existantes. Cependant, cette approche apporte une preuve du concept intéressante à travers le domaine applicatif des *BTFs*.

Pour ce type de fonctions d'apparence, le ratio entre la qualité des résultats et la taille des données est encore à améliorer. En effet, nous utilisons ici uniquement une approche par quantification et seuillage des coefficients d'ondelettes alors que des méthodes d'approximation polynomiales par exemple pourraient être utilisées conjointement pour améliorer les résultats quantitatifs. Par ailleurs, la carte de domaine ajoute à la représentation un coût de stockage non négligeable qui représente une limitation dans notre méthode.

Pour chaque brique de la méthode, les idées sur les approches futures sont nombreuses. Ainsi, pour la transformation multi-échelle de la *BTF*, nous proposons ici un schéma simplifié de transformation sur les dimensions angulaires qui est un compromis entre rapidité d'exécution et efficacité de décorrélation inter-directionnelle. Lors de nos expérimentations, une approche par fonctions à base radiale sphériques (SRBF) a été développée de manière à disposer d'un schéma d'interpolation continue sur les dimensions angulaires. Ce type de schéma est très efficace pour représenter les haute-fréquences dans le domaine directionnel car un centroïde SRBF est assigné à chaque échantillon directionnel. La complexité des calculs de reconstruction de cette interpolation est trop importante pour être utilisée en l'état car définie de manière globale. Cependant, cette approche permet d'utiliser de manière efficace les ondelettes sphériques pour déterminer les directions les plus importantes dans une base multi-échelle. Ainsi, nous envisageaons d'utiliser ces deux domaines pour définir un ensemble d'échantillons directionnels multi-échelles pour mettre en place une base d'approximation adaptative SRBF multi-échelle.

La quantification est ici définie suivant deux aspects : une métrique de qualité et une transformation permettant de stocker des réels par des entiers. La transformation adaptative des coefficients par une fonction de déformation semble être une bonne approche pour diriger la quantification. Cependant, une quantification sur un nombre variable de bits doit être analysée et comparée pour valider cette approche. Les tests effectués avec une quantification de 8 bits par coefficients sont déjà prometteur car la différence visuelle entre les données quantifiées et les réels est nulle pour une erreur relative globale de 1%. La métrique de qualité est basée sur l'erreur d'approximation quadratique globale engendrée par la suppression de coefficients d'ondelettes dans la représentation. Cette métrique nous donne une idée sur la qualité de la représentation mais ne prend pas en compte des facteurs perceptuels ou une importance par pixel. En plus de quantifier l'erreur introduite dans la représentation, l'aspect structurel de chaque image est important. En effet, le défaut majeur d'un seuillage des coefficients dans une transformée en ondelettes est l'apparition très rapide d'un effet de bloc sur l'image reconstruite dû à un seuillage trop important des haute-fréquences. Pour détecter cet effet, une analyse structurelle de l'image est possible grâce à des métriques définies par pixel comme la mesure de similarité structurelle (SSIM) [WD04] récemment développée pour des signaux en ondelettes [SWG+09]. Cette approche reviendrait à déterminer par coefficient d'ondelettes son degré d'importance dans la structure de l'image et donc, permettrait d'avoir une analyse plus fine sur la suppression des coefficients. De plus, vu que le stockage des coefficients est réalisé de manière atomique, un coefficient structurel important peut être conservé sans difficulté dans la représentation en le renseignant comme significatif dans la carte du domaine.

Le hachage des coefficients et la carte du domaine sont les points les plus importants dans cette démarche car les performances lors du rendu et les résultats quantitatifs dépendent de leur efficacité. La carte du domaine regroupe les configurations des branches des arbres de décomposition en ondelettes de chaque texture où les configurations sont factorisées afin extraire celles qui sont les plus probables dans un voisinage. Le même processus peut être réalisé à l'aide des filtres de *Bloom* [Blo70], une structure de données probabiliste qui permet de tester si un élément appartient à un ensemble. Le principe consiste à utiliser une carte de n bits et une famille de m fonctions de hachage transformant les coordonnées d'un élément en coordonnées de la carte de bits. La carte de bits est initialisée à 0 et à chaque élément non-nul d'un ensemble les m images de ses coordonnées par les fonctions de hachage sont assignés à 1. Ainsi, un élément est absent du domaine si un de ses m bits correspondant dans la table de hachage est à 0. Cette approche génère un nombre de faux-positifs dont la probabilité est connue en fonction de m et n. Cette technique nous semble donc intéressante pour définir une carte de domaine dont le nombre de faux-positifs dépend de la taille et de la complexité de cette carte. Pour la construction de la table de hachage, l'op-

timisation apportée par l'utilisation d'une famille de fonctions de hachage permet de stocker d'une manière très compacte les coefficients d'ondelettes. Cependant, nous n'utilisons pas ici les spécificités d'une décomposition en ondelettes comme dans la méthode du codage *Zero-Tree* par exemple. Ainsi, un hachage multi-échelle des coefficients peut être avantageux pour fournir une représentation progressive à travers la table de hachage elle-même. Par ailleurs, une cohérence spatiale plus forte doit être mise en place sur les différentes tables pour permettre une cohérence de cache accrue sur le matériel graphique lors du rendu pour améliorer les performances.

L'approche globale reste encore ici expérimentale mais, grâce aux travaux futurs envisagés, semble être une bonne alternative aux méthodes de codage *Zero-Tree* avec un support pour le matériel graphique. Ces travaux n'ont pas encore été publiés mais seront présentés dans les conférences à venir comme une méthode généraliste pour la compression de signaux d'ondelettes en n dimensions.

Conclusion

Résumé des contributions

Dans ce manuscrit, nous nous sommes intéressés au préalable à la définition de l'apparence d'un objet soumis à un éclairement incident suivant différents points de vue. Les conclusions de cette étude nous ont amené à focaliser nos travaux sur des fonctions d'apparence bien particulières, les « Bidirectional Texture Functions », dont le potentiel pour représenter une multitude d'effets physiques est une motivation stimulante pour mettre en avant ce type de représentation. Nous avons étudié les différents domaines liés à ce type de matériaux en effectuant une synthèse précise et argumentée de l'ensemble des travaux sur l'acquisition et la compression des *BTFs*.

Après analyse des limitations des travaux existants, nous avons présenté une transformation multi-échelle des *BTFs* qui, à ce jour, n'a jamais été développée sur l'ensemble des dimensions du matériau. Les outils utilisés pour une telle transformation des données sont les ondelettes pour lesquelles nous avons étudié, dans un premier temps, la cohérence de leur utilisation sur les *BTFs*. Ces expérimentations nous ont permis de définir un nouveau modèle de compression pour le stockage de *BTFs* dont les propriétés de scalabilité et de transmission progressive en font un outil bien plus efficace que les méthodes actuelles de stockage sur les bases de données de *BTFs* publiques.

Par la suite, nous avons continué nos travaux sur une nouvelle représentation multi-échelle de *BTFs* intitulée « arbre d'ondelettes polynomial ». Cette représentation révèle deux aspects particulièrement intéressants. D'une part, nous avons montré que l'expressivité d'un polynôme pour approximer des fonctions d'apparence peut être étendue en choisissant un domaine de paramétrisation plus opportun que ceux des méthodes classiques de la littérature. Nous avons pu ainsi réduire la quantité d'erreurs liées à l'approximation polynomiale. D'autre part, nous avons mis en place une méthode

multi-échelle pour exprimer de manière progressive un ensemble de textures en utilisant une transformation par paquet d'ondelettes. Ce type d'approche a déjà été utilisée dans le cadre de la compression d'images mais, à notre connaissance, n'a jamais permis de mettre en place une représentation compatible avec les processeurs graphiques programmables telle que nous l'avons exposée. La conjonction de ces deux contributions font que cette méthode permet de compresser d'une manière très efficace une *BTFs* tout en conservant le réalisme du matériau.

Dans le dernier chapitre, nous avons mis en place une méthode générique de compression de signaux en ondelettes que nous considérons comme la base d'un nouveau standard de compression. Cette méthode a été testée sur les *BTFs* pour lesquelles nous avons proposé une transformation simplifiée de l'espace de directions qui permet une décorélation des données angulaires sans pénaliser les performances lors de la reconstruction. Ensuite, nous proposons une méthode adaptative de quantification des coefficients d'ondelettes basée sur un critère de qualité. La fonction de quantification est définie paramétriquement ce qui permet une plus grande flexibilité de l'espace de quantification en fonction de la distribution des données. Enfin, le hachage des coefficients d'ondelettes, jamais proposé à ce jour, permet un stockage dense des données creuses exprimées dans l'espace ondelettes dont la recherche d'un coefficient par cette structure de données est réalisée en temps constant.

Perspectives

Compression

En premier lieu, nos travaux futurs vont s'orienter vers un développement plus approfondi des travaux présentés dans le Chapitre 6 sur le hachage des coefficients d'ondelettes. A partir des directions futures exposées précédemment, le premier objectif du développement d'une telle méthode est de soumettre cette approche à l'avis d'un comité de lecture dans des conférences internationales pour, ensuite, prétendre à une normalisation de ce procédé dans des consortiums tel que MPEG-4. Ces travaux seront généralisés à d'autres types de données graphiques tels que les maillages 3D, les cartes d'environnement, les cartes de normales où tout autre attribut participant au rendu d'une scène capable d'être exprimé dans une base d'ondelettes.

Perception

Dans un cadre plus général, nous envisageons de prendre en compte des facteurs perceptuels dans nos méthodes. En effet, dans toutes les approches proposées dans ce manuscrit, l'erreur quadratique est la métrique privilégiée

pour juger de la qualité de nos représentations. Cependant, bien que nécessaire, cette approche n'est pas suffisante pour conjecturer sur l'aspect perceptuel de l'erreur introduite. D'après une étude sur les différentes solutions proposées dans l'état de l'art, nous avons déterminé que les approches structurelles semblent être une méthode pertinente pour analyser de manière plus fine les gênes perceptuelles occasionnées par la dégradation d'un signal visuel. Dans le cadre des *BTFs*, l'analyse structurelle est particulièrement intéressante car elle peut être effectuée à la fois sur les dimensions spatiales pour préserver les spécificités méso-structurelles mais aussi sur les dimensions angulaires de manière, par exemple, à extraire les effets engendrés par la parallaxe entre les différents points de vue.

Acquisition

Nous considérons les *BTFs* comme une des meilleures approches actuelles pour des fonctions d'apparence de part le réalisme de ces données et les nombreux effets physiques qu'elles peuvent représenter. Cependant, les problématiques majeures sur ce type de matériau sont liées à l'acquisition. En effet, les dispositifs d'acquisition actuels nécessitent de nombreux post-traitements pour exprimer l'ensemble des textures de manière uniforme tant dans la résolution que dans l'orientation de la normale à la texture. Ces traitements engendrent des erreurs d'alignement des données ce qui n'est pas trop important lorsque le matériau acquis possède de forts effets de parallaxe mais qui sont visibles pour des matériaux de faibles élévations. De nouveaux systèmes d'acquisition doivent donc être mis en place pour limiter les post-traitements en utilisant un système de prisme, par exemple, pour rediriger le signal acquis dans la configuration voulue. De plus, la méso-structure d'un matériau peut être mesurée lors de l'acquisition grâce à une combinaison entre un scanner 3D et un périphérique de capture de l'apparence. Ces informations géométriques permettraient de développer de nouvelles représentations bien plus efficaces car, par cette approche, les effets de parallaxe pourraient être gérés d'une manière plus efficace.

Bibliographie

[AP07] Michael Ashikhmin and Simon Premoze. Distribution-based BRDFs. Technical Report http://www.cs.utah.edu/~premoze/dbrdf/, University of Utah, 2007.

[APS00] M. Ashikhmin, S. Premoze, and P. Shirley. A microfacet-based BRDF generator. In *ACM SIGGRAPH '00*, pages 65–74, 2000.

[AS00] Michael Ashikhmin and Peter Shirley. An anisotropic phong brdf model. *J. Graph. Tools*, 5(2) :25–32, 2000.

[ASA+09] Dan A. Alcantara, Andrei Sharf, Fatemeh Abbasinejad, Shubhabrata Sengupta, Michael Mitzenmacher, John D. Owens, and Nina Amenta. Real-time parallel hashing on the gpu. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2009)*, 28(5), December 2009.

[BA83] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4 :532–540, 1983.

[BBGS08] Jerome Baril, Tamy Boubekeur, Patrick Gioia, and Christophe Schlick. Polynomial wavelet trees for bidirectional texture functions. In *ACM Siggraph 2008 - Talk Program*. ACM, aug 2008.

[BBS94] Deborah F. Berman, Jason T. Bartell, and David H. Salesin. Multiresolution painting and compositing. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 85–90, New York, NY, USA, 1994. ACM.

[BG08a] Jérôme Baril and Patrick Gioia. Method for encoding data representative of a multi-dimensional texture, encoding device and corresponding decoding method and device, signal and software. *Patent International Application No. PCT/FR2008/050161*, 2008.

[BG08b] Jérôme Baril and Patrick Gioia. Procédé de compression de matériaux complexes et méthodes de rendu associées. *No. 09 50522*, 2008.

[Bli77] James F. Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH '77*, pages 192–198, 1977.

[Bli78] James F. Blinn. Simulation of wrinkled surfaces. In *SIGGRAPH '78 : Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 286–292, New York, NY, USA, 1978. ACM Press.

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7) :422–426, 1970.

[Bou08] Mike Boulton. Using wavelets with current and future hardware. In *SIGGRAPH '08 : ACM SIGGRAPH 2008 classes*, pages 102–132, New York, NY, USA, 2008. ACM.

[BR04] Shaun Bangay and Judith D. Radloff. Kaleidoscope configurations for reflectance measurement. In *AFRIGRAPH '04 : Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 161–170, New York, NY, USA, 2004. ACM.

[BTFa] BTF Database Bonn. http ://btf.cs.uni-bonn.de.

[BTFb] BTFShop. http ://btfshop.sourceforge.net/.

[CBP07] Luc Claustres, Loïc Barthe, and Mathias Paulin. Wavelet encoding of brdfs for real-time rendering. In *GI '07 : Proceedings of Graphics Interface 2007*, pages 169–176, New York, NY, USA, 2007. ACM Press.

[CCC87] Robert L. Cook, Loren Carpenter, and Edwin Catmull. The reyes image rendering architecture. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 95–102, New York, NY, USA, 1987. ACM.

[CDF92] A. Cohen, Ingrid Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45(5) :485–560, 1992.

[CJAMJ05] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet importance sampling : efficiently evaluating products of complex functions. *ACM Trans. Graph.*, 24(3) :1166–1175, 2005.

[CLM] CLML. Cornell Light Measurement Laboratory.

[CMS87] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 273–281, New York, NY, USA, 1987. ACM Press.

[Coo84] Robert L. Cook. Shade trees. *SIGGRAPH Comput. Graph.*, 18(3) :223–231, 1984.

[CPD+96] Andrew Certain, Jovan Popovic, Tony DeRose, Tom Duchamp, David Salesin, and Werner Stuetzle. Interactive multiresolution surface viewing. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 91–98, New York, NY, USA, 1996. ACM.

[CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1) :7–24, 1982.

[Cud] Cuda. CUDA. http ://developer.nvidia.com/page/home.html.

[Cur] Curet. Columbia-Utrecht Reflectance and Texture Database. http ://www1.cs.columbia.edu/CAVE/software/curet/.

[Dau92] Ingrid Daubechies. *Ten lectures on wavelets.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[DLHS01] Katja Daubert, Hendrik P. A. Lensch, Wolfgang Heidrich, and Hans-Peter Seidel. Efficient cloth modeling and rendering. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 63–70, London, UK, 2001. Springer-Verlag.

[DNGK97] Kristin J. Dana, Shree K. Nayar, Bram Van Ginneken, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. In *CVPR '97 : Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 151, Washington, DC, USA, 1997. IEEE Computer Society.

[DR08] Julie Dorsey and Holly Rushmeier. Light and materials in virtual cities. In *SIGGRAPH 2008 Course Notes*, 2008.

[DvGNK99] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1) :1–34, 1999.

[EBJ+06] Dave Edwards, Solomon Boulos, Jared Johnson, Peter Shirley, Michael Ashikhmin, Michael Stark, and Chris Wyman. The halfway vector disk for BRDF modeling. *ACM Trans. Graph.*, 25(1) :1–18, 2006.

[EDD+95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH '95 : Proceedings of*

the 22nd annual conference on Computer graphics and interactive techniques, pages 173–182, New York, NY, USA, 1995. ACM.

[FCGH08] J. Filip, M.J. Chantler, P.R. Green, and M. Haindl. A psychophysically validated metric for bidirectional texture data reduction. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008)*, 27(5) :138, December 2008.

[FCH08] J. Filip, M.J. Chantler, and M. Haindl. On optimal resampling of view and illumination dependent textures. In *Fifth Symposium on Applied Perception in Graphics and Visualization*, pages 131–134, August 2008.

[FCH09] J. Filip, M.J. Chantler, and M. Haindl. On uniform resampling and gaze analysis of bidirectional texture functions. *To appear in ACM Transactions on Applied Perception*, 6 :15, Septrember 2009.

[FH04] Jiri Filip and Michal Haindl. Non-linear reflectance model for bidirectional texture function synthesis. In *ICPR '04 : Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1*, pages 80–83, Washington, DC, USA, 2004. IEEE Computer Society.

[FH05] J. Filip and M. Haindl. Efficient image based bidirectional texture function model. In M. Chantler and O. Drbohlav, editors, *Texture 2005 : Proceedings of 4th Internatinal Workshop on Texture Analysis and Synthesis*, pages 7–12, Edinburgh, October 2005. Heriot-Watt University.

[FH07] J. Filip and M. Haindl. BTF modelling using BRDF texels. *International Journal of Computer Mathematics (IJCM)*, 84(9) :1267–1283, September 2007.

[FH08] Jiří Filip and Michal Haindl. Bidirectional texture function modeling : A state of the art survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(1), 2008.

[FKIS02] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi. Appearance based object modeling using texture database : acquisition, compression and rendering. In *EGRW '02 : Proceedings of the 13th Eurographics workshop on Rendering*, pages 257–266, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[Foo96] Sing-Choong Foo. A gonioreflectometer for measuring the bidirectional reflectance of material for use in illumination computation. Master's thesis, Cornell University, 1996.

[GZ05] Michael Garland and Yuan Zhou. Quadric-based simplification in any dimension. *ACM Trans. Graph.*, 24(2) :209–239, 2005.

[Had09] Julien Hadim. *Etude en vue de la multirésolution de l'apparence*. PhD thesis, LaBRI- Université de Bordeaux 1, 2009.

[Hai07] Jiri Haindl, Michal ; Filip. Extreme compression and modeling of bidirectional texture function. *Transactions on Pattern Analysis and Machine Intelligence*, 29(10) :1859–1865, 2007.

[HCFE04] E. Hayman, B. Caputo, M. Fritz, and J.O. Eklundh. On the significance of real-world conditions for material classification. pages Vol IV : 253–266, 2004.

[HNZ+05] Vlastimil Havran, Attila Neumann, Georg Zotti, Werner Purgathofer, and Hans-Peter Seidel. On cross-validation and resampling of brdf data measurements. In *SCCG '05 : Proceedings of the 21st spring conference on Computer graphics*, pages 161–168, New York, NY, USA, 2005. ACM.

[Hop96] Hugues Hoppe. Progressive meshes. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99–108, New York, NY, USA, 1996. ACM.

[HP03] Jefferson Y. Han and Ken Perlin. Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Trans. Graph.*, 22(3) :741–748, 2003.

[HPB06] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Direct-to-indirect transfer for cinematic relighting. In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Papers*, pages 1089–109<7, New York, NY, USA, 2006. ACM.

[HS98] Wolfgang Heidrich and Hans-Peter Seidel. View-independent environment maps. In *HWWS '98*, pages 39–ff., New York, NY, USA, 1998. ACM.

[HTSG91] Xiao D. He, Kenneth E. Torrance, FranÃ§ois X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. In *ACM SIGGRAPH '91*, pages 175–186, 1991.

[JMLH01] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518, New York, NY, USA, 2001. ACM Press.

[JPE00] JPEG. Part 1 Core coding system. *ISO/IEC 15444-1*, 2000.

[KBD07] Jan Kautz, Solomon Boulos, and Frédo Durand. Interactive editing and modeling of bidirectional texture functions. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 53, New York, NY, USA, 2007. ACM.

[KDS96] Jan J. Koenderink, Andrea J. van Doorn, and Marigo Stavridi. Bidirectional Reflection Distribution Function Expressed in Terms of Surface Scattering Modes. In *Proc. European Conference on Computer Vision-Volume II*, 1996.

[KM99] Jan Kautz and Michael D. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Proc. Eurographics Workshop on Rendering*, pages 247–260. Springer, June 1999.

[KMBK03] M. L. Koudelka, S. Magda, P. N. Belhumeu, and D. J. Kriegman. Acquisition, compression and synthesis of bidirectional texture functions. In *3rd International Workshop on Texture Analysis and Synthesis*, 2003.

[KTHS06] Janne Kontkanen, Emmanuel Turquin, Nicolas Holzschuch, and François Sillion. Wavelet radiance transport for interactive indirect lighting. In Wolfgang Heidrich and Thomas Akenine-Möller, editors, *Rendering Techniques 2006 (Eurographics Symposium on Rendering)*. Eurographics, jun 2006.

[LBAD+06] Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Inverse shade trees for non-parametric material representation and editing. *ACM Trans. Graph.*, 25(3) :735–745, 2006.

[LDW97] Michael Lounsbery, Tony D. DeRose, and Joe Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1) :34–73, 1997.

[LF08] Christian Lessig and Eugene Fiume. Soho : Orthogonal and symmetric haar wavelets on the sphere. *ACM Trans. Graph.*, 27(1) :1–11, 2008.

[LFTG97] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[LH96] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, New York, NY, USA, 1996. ACM.

[LH06] Sylvain Lefebvre and Hugues Hoppe. Perfect spatial hashing. In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Papers*, pages 579–588, New York, NY, USA, 2006. ACM.

[LH07] Sylvain Lefebvre and Hugues Hoppe. Compressed random-access trees for spatially coherent data. In *Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering)*. Eurographics, 2007.

[Llo82] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2) :129–137, January 1982.

[LRR04] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient BRDF importance sampling using a factored representation. *ACM Trans. Graph.*, 23(3), 2004.

[LSP05] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8) :1265–1278, 2005. Student Member-Svetlana Lazebnik and Senior Member-Cordelia Schmid and Fellow-Jean Ponce.

[Lys09] D. Lyssi. A reflectometer setup for spectral btf measurement. To appear in Proceedings of CESCG, 2009.

[MAA01] Michael D. McCool, Jason Ang, and Anis Ahmad. Homomorphic factorization of BRDFs for high-performance rendering. In *ACM SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 171–178. ACM Press, 2001.

[Mal89] S. G. Mallat. *A Theory for Multiresolution Signal Decomposition : The Wavelet Representation*, volume 11. IEEE Computer Society, Washington, DC, USA, 1989.

[MBK05] G. Müller, G. H. Bendels, and R. Klein. Rapid synchronous acquisition of geometry and btf for cultural heritage artefacts. In *The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*. Eurographics Association, Eurographics Association, November 2005.

[McA02] David K. McAllister. *A Generalized Surface Appearance Representation for Computer Graphics*. PhD thesis, UNC Chapel Hill, 2002.

[MCC+04] Wan-Chun Ma, Sung-Hsiang Chao, Bing-Yu Chen, Chun-Fa Chang, Ming Ouhyoung, and Tomoyuki Nishita. An efficient representation of complex materials for real-time rendering. pages 150–153, 2004.

[MCT+05] Wan-Chun Ma, Sung-Hsiang Chao, Yu-Ting Tseng, Yung-Yu Chuang, Chun-Fa Chang, Bing-Yu Chen, and Ming Ouhyoung. Level-of-detail representation of bidirectional texture functions for real-time rendering. In *SI3D '05 : Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 187–194, New York, NY, USA, 2005. ACM Press.

[MGW01] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial texture maps. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 519–528, New York, NY, USA, 2001. ACM.

[MLH02] David K. McAllister, Anselmo Lastra, and Wolfgang Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. In *HWWS '02 : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 79–88, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[MMK03a] J. Meseth, G. Müller, and R. Klein. Preserving realism in real-time rendering of bidirectional texture functions. In *OpenSG Symposium 2003*, pages 89–96. Eurographics Association, Switzerland, April 2003.

[MMK03b] G. Müller, J. Meseth, and R. Klein. Compression and real-time rendering of measured btfs using local pca. In T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, E. Steinbach, and R. Westermann, editors, *Vision, Modeling and Visualisation 2003*, pages 271–280. Akademische Verlagsgesellschaft Aka GmbH, Berlin, November 2003.

[MMK04a] J. Meseth, G. Müller, and R. Klein. Reflectance field based real-time, high-quality rendering of bidirectional texture functions. *Computers and Graphics*, 28(1) :103–112, February 2004.

[MMK04b] G. Müller, J. Meseth, and R. Klein. Fast environmental lighting for local-pca encoded btfs. In *Computer Graphics International, 2004. Proceedings*, pages 198–205, 2004.

[MMS+04] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein. Acquisition, synthesis and rendering of bidirectional texture functions. In Christophe Schlick and Werner Purgathofer, editors, *Eurographics 2004, State of the Art Reports*, pages 69–94. INRIA and Eurographics Association, September 2004.

[MPBM03] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3) :759–769, July 2003.

[MSK07] Gero Müller, Ralf Sarlette, and Reinhard Klein. Procedural editing of bidirectional texture functions. In J. Kautz and S. Pattanaik, editors, *Eurographics Symposium on Rendering 2007*. The Eurographics Association, June 2007.

[MWAM05] Stephen R. Marschner, Stephen H. Westin, Adam Arbree, and Jonathan T. Moon. Measuring and modeling the appearance of finished wood. *ACM Trans. Graph.*, 24(3) :727–734, 2005.

[MZD05] Wojciech Matusik, Matthias Zwicker, and Fr&#233 ;do Durand. Texture design using a simplicial complex of morphable textures. *ACM Trans. Graph.*, 24(3) :787–794, 2005.

[ND06] Addy Ngan and Frédo Durand. Statistical Acquisition of Texture Appearance. In Tomas Akenine-Möller and Wolfgang Heidrich, editors, *Proc. Eurographics Symposium on Rendering*, pages 31–40, Nicosia, Cyprus, 2006. Eurographics Association.

[NDM05] A. Ngan, F. Durand, and W. Matusik. Experimental Analysis of BRDF Models. In *Proc. Eurographics Symposium on Rendering*, pages 117–226, 2005.

[Nic70] F. E. Nicodemus. Reflectance nomenclature and directional reflectance and emissivity. *Appl. Opt.*, 9 :1474–1475, jun 1970.

[NRH04] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph.*, 23(3) :477–487, 2004.

[NSI01] Ko Nishino, Yoichi Sato, and Katsushi Ikeuchi. Eigen-texture method : Appearance compression and synthesis based on a 3d model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11) :1257–1265, 2001.

[OS95] Peter Schr Oder and Wim Sweldens. Spherical wavelets : Texture processing. In *Rendering Techniques '95*, 1995.

[PAB00] Christophe Parisot, Marc Antonini, and Michel Barlaud. Ebwic : A low complexity and efficient rate constrained wavelet image coder. In *ICIP*, 2000.

[PFTV88] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical recipes in C : the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1988.

[Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6) :311–317, 1975.

[PL07] Fabio Pellacini and Jason Lawrence. Appwand : editing measured materials using appearance-driven optimization. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 54, New York, NY, USA, 2007. ACM.

[PM05] Gabriel Peyré and Stéphane Mallat. Surface compression with geometric bandelets. *ACM Trans. Graph.*, 24(3) :601–608, 2005.

[PO06] Fabio Policarpo and Manuel M. Oliveira. Relief mapping of non-height-field surface details. In *I3D '06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 55–62, New York, NY, USA, 2006. ACM.

[POC05] Fábio Policarpo, Manuel M. Oliveira, and ao L. D. Comba, Jo˙ Real-time relief mapping on arbitrary polygonal surfaces. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 935–935, New York, NY, USA, 2005. ACM.

[PVL+05] Fabio Pellacini, Kiril Vidimče, Aaron Lefohn, Alex Mohr, Mark Leone, and John Warren. Lpics : a hybrid hardware-accelerated relighting engine for computer cinematography. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 464–470, New York, NY, USA, 2005. ACM.

[PWS06] Drineas P., Mahoney M. W., and Muthukrishnan S. Polynomial time algorithm for column-row based relative-error low-rank matrix approximation. Technical report, DIMACS, 2006.

[RH04] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for reflection. *ACM Trans. Graph.*, 23(4) :1004–1042, 2004.

[RK09] Roland Ruiters and Reinhard Klein. Btf compression via sparse tensor decomposition. *Computer Graphics Forum*, 28(4), jul 2009.

[RKB05] Ravi Ramamoorthi, Melissa Koudelka, and Peter Belhumeur. A fourier theory for cast shadows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(2) :288–295, 2005.

[RRK09] Roland Ruiters, Martin Rump, and Reinhard Klein. Parallelized matrix factorization for fast btf compression. In *Eurographics Symposium on Parallel Graphics and Visualization*, pages 25–32, 2009.

[Rus98] Szymon Rusinkiewicz. A new change of variables for efficient BRDF representation. In *Rendering Techniques (Proc. Eurographics Workshop on Rendering)*, June 1998.

[SBLD03] Frank Suykens, Karl Berge, Ares Lagae, and Philip Dutre. Interactive rendering with bidirectional texture functions. *Computer Graphics Forum*, 22(3) :463–472, 2003.

[Sch93] Christophe Schlick. A customizable reflectance model for everyday rendering. In *In Fourth Eurographics Workshop on Rendering*, pages 73–83, 1993.

[Sch94a] Christophe Schlick. A Survey of Shading and Reflectance Models. *Comp. Graph. Forum*, 13(2), 1994.

[Sch94b] Christophe Schlick. An Inexpensive BRDF Model for Physically-Based Rendering. *Computer Graphics Forum*, 13(3) :233–246, 1994.

[Sch94c] Christophe Schlick. A fast alternative to phong's specular model. pages 385–387, 1994.

[Sch94d] Christophe Schlick. *Graphics Gems V*, chapter A Fast Alternative to Phong's Specular Model, pages 385–384. Morgan Kauffman, 1994.

[Sch94e] Christophe Schlick. Quantization techniques for visualization of high dynamic range pictures. pages 7–18, 1994.

[SDS95a] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics : A primer, part 1. *IEEE Comput. Graph. Appl.*, 15(3) :76–84, 1995.

[SDS95b] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics : A primer, part 2. *IEEE Comput. Graph. Appl.*, 15(4) :75–85, 1995.

[SDS96] Eric J. Stollnitz, Tony D. Derose, and David H. Salesin. *Wavelets for computer graphics : theory and applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[Sha93] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on*, 41(12) :3445–3462, 1993.

[SKP05] Musawir A. Shah, Jaakko Kontinnen, and Sumanta Pattanaik. Real-time rendering of realistic-looking grass. In *GRAPHITE '05 : Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 77–82, New York, NY, USA, 2005. ACM Press.

[SMSA01] P. Shirley, S. Marschner, J. Stam, and M. Ashikhmin. State of the art in modeling and measuring of surface reflection. In *SIGGRAPH 2001 Course Notes*, 2001.

[SP96] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *CirSysVideo*, 6(3) :243–250, June 1996.

[SS95] Peter Schröder and Wim Sweldens. Spherical wavelets : efficiently representing functions on the sphere. In *SIGGRAPH '95 : Proc. 22nd annual conference on Computer graphics and interactive techniques*, pages 161–172, 1995.

[SSK03] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Efficient and realistic visualization of cloth. In *EGSR '03 : Proceedings of the 14th Eurographics Symposium on Rendering*, pages 167–177, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[Sta99] Jos Stam. Diffraction shaders. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 101–110, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[Swe96] W. Sweldens. The lifting scheme : A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2) :186–200, 1996.

[Swe97] W. Sweldens. The lifting scheme : A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2) :511–546, 1997.

[SWG+09] Mehul P. Sampat, Zhou Wang, Shalini Gupta, Alan C. Bovik, and Mia K. Markey. Complex wavelet structural similarity : A new image similarity index. *Image Processing, IEEE Transactions on*, 18(11) :2385–2401, June 2009.

[Tat06] Natalya Tatarchuk. Dynamic parallax occlusion mapping with approximate soft shadows. In *I3D '06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 63–69, New York, NY, USA, 2006. ACM.

[Tau00] David S. Taubman. High performance scalable image compression with ebcot. *IEEE Transactions on Image Processing*, 9(7) :1158–1170, 2000.

[TP08] Christian Tenllado and Luis Pi nuel. Parallel implementation of the 2d discrete wavelet transform on graphics processing units : Filter bank versus lifting. *IEEE Trans. Parallel Distrib. Syst.*, 19(3) :299–310, 2008. Member-Setoain, Javier and Member-Prieto, Manuel and Senior Member-Tirado, Francisco.

[TS67] K. Torrance and E. Sparrow. Polarization, directional distribution, and off-specular peak phenomena in light reflected from roughened surfaces. *J. Opt. Soc. Am.*, 57(9) :1105–1114, 1967.

[VT04] M. Alex O. Vasilescu and Demetri Terzopoulos. Tensortextures : multilinear image-based rendering. *ACM Trans. Graph.*, 23(3) :336–342, 2004.

[War92] Gregory J. Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 265–272, New York, NY, USA, 1992. ACM Press.

[WAT92] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. In *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 255–264, New York, NY, USA, 1992. ACM Press.

[WD04] Jing Wang and Kristin J. Dana. Hybrid textons : Modeling surfaces with reflectance and geometry. In *CVPR (1)*, pages 372–378, 2004.

[WD06] Jing Wang and Kristin J. Dana. Relief texture from specularities. volume 28, pages 446–457, Washington, DC, USA, 2006. IEEE Computer Society.

[WDR09] Hongzhi Wu, Julie Dorsey, and Holly Rushmeier. Characteristic point maps. In *Eurographics Symposium on Rendering*. Eurographics, 2009.

[Wer] Bernard Werber. L'arbre des possibles. http ://www.arbredespossibles.com.

[Wey05] Tim Weyrich. Rendering deformable surface reflectance fields. *IEEE Transactions on Visualization and Computer Graphics*, 11(1) :48–58, 2005. Senior Member-Hanspeter Pfister and Senior Member-Markus Gross.

[WHON97] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng. Image-based rendering with controllable illumination. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97*, pages 13–22, London, UK, 1997. Springer-Verlag.

[WLHW07] T.T. Wong, C.S. Leung, P.A. Heng, and J.Q. Wang. Discrete wavelet transform on consumer-level graphics hardware. 9(3) :668–673, April 2007.

[WWHL07] R. Peter Weistroffer, Kristen R. Walcott, Greg Humphreys, and Jason Lawrence. Efficient Basis Decomposition for Scattered Reflectance Data. In *Proc. Eurographics Symposium on Rendering*, 2007.

[WWS+05a] Hongcheng Wang, Qing Wu, Lin Shi, Yizhou Yu, and Narendra Ahuja. Out-of-core tensor approximation of multi-dimensional matrices of visual data. *ACM Trans. Graph.*, 24(3) :527–535, 2005.

[WWS+05b] Hongcheng Wang, Qing Wu, Lin Shi, Yizhou Yu, and Narendra Ahuja. Out-of-core tensor approximation of multi-dimensional matrices of visual data, 2005.

[WWT+03] Lifeng Wang, Xi Wang, Xin Tong, Stephen Lin, Shimin Hu, Baining Guo, and Heung-Yeung Shum. View-dependent displacement mapping. In *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers*, pages 334–339, New York, NY, USA, 2003. ACM.

[YC05] Yizhou Yu and Johnny T. Chang. Shadow graphs and 3d texture reconstruction. *Int. J. Comput. Vision*, 62(1-2) :35–60, 2005.

[ZDW+05] Kun Zhou, Peng Du, Lifeng Wang, Jiaoying Shi, Baining Guo, and Heung-Yeung Shum. Decorating surfaces with bidirectional texture functions. *IEEE Transactions on Visualization and Computer Graphics*, 11(5) :519–528, 2005. Member-Yasuyuki Matsushita.

[ZERB05] Todd Zickler, Sebastian Enrique, Ravi Ramamoorthi, and Peter Belhumeur. Reflectance sharing : Image-based rendering from a sparse set of images. In *Proc. Eurographics Symposium on Rendering*, pages 253–265, 2005.