



HAL
open science

Traitement conjoint de la géométrie et de la radiance d'objets 3D numérisés

Kenneth Vanhoey

► **To cite this version:**

Kenneth Vanhoey. Traitement conjoint de la géométrie et de la radiance d'objets 3D numérisés. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université de Strasbourg, 2014. Français. NNT : 2014STRAD005 . tel-01199404

HAL Id: tel-01199404

<https://theses.hal.science/tel-01199404>

Submitted on 15 Sep 2015

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

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

N° d'ordre: 2315



THÈSE DE DOCTORAT

présentée pour obtenir le grade de
Docteur de l'université de Strasbourg
Mention : **Informatique**

par

Kenneth VANHOEY

Traitement conjoint de la géométrie et de la radiance d'objets 3D numérisés

Thèse soutenue publiquement le **18 février 2014**

Membres du jury :

M. Kadi BOUATOUCH *Rapporteur*
Professeur à l'université de Rennes 1

M. Bruno LÉVY *Rapporteur*
Directeur de recherche au centre INRIA Nancy-Grand Est

M. Mateu SBERT *Examineur*
Professeur à l'université de Gérone, Espagne

M. Basile SAUVAGE *Examineur et encadrant*
Maître de conférences à l'université de Strasbourg

M. Jean-Michel DISCHLER *Directeur de thèse*
Professeur à l'université de Strasbourg

Remerciements

Parce que sans toutes les personnes que je m'apprête à citer, et sans doute beaucoup d'autres encore, je n'aurais jamais pu débiter et encore moins terminer ce projet doctoral, je souhaite vivement débiter ce document en leur exprimant mes remerciements les plus sincères. À ce titre, je souhaite remercier toute personne qui a pu, par ses critiques, son soutien ou son simple intérêt en mon projet, contribuer à mon doctorat et à mon vécu de celui-ci.

Avant tout, je me dois de remercier mes encadrants, qui, dès le début, ont œuvré pour que j'ai accès à ce doctorat. Basile et Jean-Michel ont ensuite dirigé ce projet de façon très complémentaire : le premier en étant souvent disponible, en me poussant un peu quand il le fallait et en étant toujours d'excellent conseil que ce soit pour des pratiques de recherche, d'enseignement ou d'organisation du travail en général ; le second en orientant et supervisant les travaux plus globalement et en proposant des critiques constructives de plus haut niveau. Le tout toujours dans une ambiance agréable professionnellement et humainement. Très sincèrement, ce fût un plaisir de travailler dans de telles conditions d'encadrement.

Basile, je souhaite en particulier t'exprimer ma gratitude pour les innombrables discussions sur la recherche, l'enseignement ou sur la vie, durant lesquelles tu m'as énormément appris. Merci d'avoir (tenté) de me protéger contre mes innombrables envies de faire trop de choses (les tâches ne manquent pas à l'université), et merci surtout d'avoir réussi à concilier ta présence avec mon autonomie.

Merci ensuite aux membres du jury et rapporteurs de ces travaux, MM. Bruno LEVY, Kadi BOUATOUCH, Mateu SBERT ainsi que Pierre GRUSSENMEYER, d'avoir accepté d'évaluer ces travaux et de m'avoir fait part de remarques pertinentes ayant conduit à l'amélioration du manuscrit final.

De nombreux éléments des travaux rapportés dans cette thèse s'appuient sur des connaissances théoriques et techniques de l'équipe en place. Je souhaite ainsi chaleureusement remercier Frédéric et Olivier pour leur disponibilité et leur participation active à ce projet de recherche. Ils ont été d'une très précieuse aide de par leur connaissance en termes de numérisation et leurs capacités techniques exceptionnelles.

Parce qu'ils m'ont attiré vers la recherche lors de mes années d'étudiant, je dois également remercier Nicolas (PASSAT) et Simon. Lorsque l'idée de faire un doctorat à pris forme, ils ont tous les deux été très disponible pour m'orienter et m'expliquer, aussi bien que possible lorsqu'on ne l'a pas vécu, ce que cela implique (malgré leurs visions des fois très divergentes à ce sujet).

Ik wil hierbij ook graag Dhr. Jacques VERHOEVEN hartelijk danken. Zijn bijdrage in mijn motivatie en leerzucht tijdens mijn laatste school –en eerste universiteits-jaren is zeker niet miniem geweest.

Plus généralement, je remercie l'équipe IGG pour le très agréable cadre de travail qu'elle propose.

Parmi les membres la composant, j'ai plus particulièrement eu l'occasion de travailler et d'interagir avec l'équipe de développement de *CGoGN*. Y contribuer modestement a d'abord, aux débuts de la librairie, été un joyeux casse-tête avant de progressivement devenir un plaisir de par les discussions animées et l'entraide permanente de Basile, Lionel, Pierre, Sylvain et Thomas.

Et pour terminer, je tiens à remercier tous les membres actuels ou passés de l'équipe, avec qui il j'ai pu me lier d'amitié ou plus simplement avec qui il fait toujours bon discuter ou passer des moments de détente au laboratoire ou ailleurs. À ce titre, je remercie, outre les personnes déjà citées, et en tentant de n'oublier personne, Alex, Alvaro, Amir, Arash, Caroline, Dominique, Étienne, Gonzalo, Guillaume, Guoliang, Jonathan, Julien, Nicolas, Noura, Pascal et Pascal, Pierre, Sabah, Thomas, Toni et Vasyl.

J'ai également eu l'occasion de collaborer avec des personnes du département d'informatique enseignement. Merci ainsi à Pierre (Tellier), Sophie, Benoît, Gabriel et Stella ainsi qu'à nouveau à Nicolas (Passat) et Basile pour leurs conseils, leur agréable collaboration et l'autonomie qu'ils m'ont rapidement accordés. Merci aussi aux étudiants qui se sont intéressés à mon projet doctoral en travaillant sur des projets y afférant, ce qui est le cas en particulier de Nicolas, Armelle et Jacques.

Et pour terminer cette section « collègues », je souhaite exprimer un grand merci tout particulier à Lionel, avec qui je partage mes journées universitaires depuis bientôt 9 ans. Il a été d'un grand soutien, déjà rien que par son écoute en cas de périodes difficiles, mais aussi par d'innombrables conseils et notamment par tout ce qu'il a pu m'apprendre. Je me souviendrais toujours de mon premier TP en licence 1 en septembre 2005, où tu étais mon voisin (inconnu pour moi) en salle machine. Tu m'as généreusement fait survivre la séance en m'expliquant cette chose bizarre qu'était la ligne de commande auquel je ne connaissais absolument rien à l'époque. Lionel, pour tout cela : merci !

Durant ces années passées à travailler sur mon projet doctoral, j'ai également eu la joie de défendre la cause des « jeunes chercheurs » au sein (et un peu au-delà) de l'université de Strasbourg. J'ai à ce titre pu rencontrer et travailler avec de nombreuses personnes intéressantes et impliquées, partenaires institutionnels ou bénévoles comme moi, qui m'ont beaucoup apporté et beaucoup appris en termes de travail d'équipe et de relations humaines. Je remercie ainsi toutes les personnes avec qui j'ai pu travailler dans les institutions universitaires ainsi que dans les associations Addal, DEHSPUS et DoXtra. J'ai particulièrement apprécié de travailler avec Anis, Antoine, Cécilia, Mike, Olivier et Simon, véritables défenseurs de la cause des doctorants qui cherchent toujours à obtenir des avancées concrètes.

Parce qu'être motivé et travailler dur ne peut, pour ma part en tout cas, que se faire en alternance régulière avec des moments de détente et de prise de recul, je souhaite remercier également les personnes qui ont contribué à cet équilibre. Il s'agit particulièrement de mes amis étudiants que j'ai pu quotidiennement côtoyer pendant longtemps (et malheureusement un peu moins récemment) et que sont Marion, Benoît, Lionel & Mélina et Dimitri & Lucie. Un grand merci également à mes amis Glynis & Jim qui, en plus d'être toujours à l'écoute et d'excellent conseil, sont les partenaires rêvés pour prendre du recul (et surtout de l'altitude). Merci également à Marie-Christine et Jean-Michel qui se sont toujours intéressés à mon devenir et à mon projet professionnel, tout en étant un exemple hors du commun de volonté et de réussite. Et merci finalement aux personnes ayant fait le déplacement pour ma soutenance et ont de ce fait embellie cette (déjà très belle) journée.

Je souhaite terminer cette partie en exprimant une pensée toute particulière pour mes plus proches. Oneindig veel dank gaat naar mijn ouders en Gill, die alle drie een buitengewoon voorbeeld vormen en mij altijd op materieel en emotioneel vlak het nodige gegund hebben om de weg van mijn keuze te volgen. Pour conclure, je dois un très chaleureux merci à Camille qui m'aura apporté au quotidien le soutien et l'apaisement qui me permet de toujours appréhender le lendemain avec confiance et motivation.

Résumé

La numérisation de la forme et de l'apparence d'objets réels est une solution intéressante dans le domaine de l'étude et de la préservation du patrimoine culturel. Les objets virtuellement reconstruits peuvent en effet être utiles dans un but d'archivage, d'étude à distance voire, dans un second temps, de restauration. À ce titre, tant la forme (géométrie) que l'apparence (photométrie) peuvent être acquises et reconstruites avec un niveau de détails variable. Pour représenter l'apparence, les champs de lumière surfaciques permettent d'encoder et de visualiser des effets directionnels (*e.g.* spécularité) induits par la réflexion de la lumière sur la surface de l'objet. Il s'agit là de fonctions 4D définissant la couleur d'un objet pour tout point de sa surface et pour toute direction d'observation autour de l'objet.

Nous nous intéressons dans cette thèse à l'acquisition, la reconstruction, la simplification et la visualisation du champ de lumière d'un objet acquis dont la géométrie est représentée par un maillage dense.

Dans un premier temps, nous analysons la littérature afin de présenter les différentes méthodes permettant d'acquérir et de représenter la photométrie, notamment le champ de lumière. Nous présentons ensuite les différentes représentations d'un champ de lumière et détaillons notamment celle que nous mettons en œuvre : le champ de lumière surfacique exploitant des fonctions 2D, dites « de radiance », localisées sur la surface.

Ensuite, nous présentons des contributions se situant dans trois parties distinctes du processus de numérisation : la reconstruction à partir d'acquisitions puis la visualisation et la simplification des données reconstruites.

Ainsi nous débutons par la reconstruction à partir d'une acquisition réelle du champ de lumière. Le processus commence par une numérisation *in situ*, tel qu'il est souvent nécessaire lorsque l'on considère des objets du patrimoine (fragiles ou de grande taille). Après acquisition et reconstruction de la géométrie, un ensemble de photos forme un échantillonnage du champ de lumière. Ces données nous permettent de reconstruire le champ de lumière continu dans l'espace des points de vue, que nous représentons sous forme de fonctions de radiance « sortante » définies sur la surface. Ces fonctions définissent la couleur d'un point de la surface en fonction de la direction d'observation. De nombreux problèmes peuvent cependant survenir lors de l'acquisition : les données peuvent être creuses ou mal échantillonnées dans l'espace des points de vue. De plus, les appareils de mesure et les algorithmes de reconstruction et de recalage introduisent généralement du bruit dans ces données. Ceci rend instable l'ajustement de fonctions de radiance aux données d'entrée. Nous proposons un algorithme permettant de créer ces fonctions en choisissant un compromis entre précision par rapport aux photos d'entrée et stabilité. De plus, nous proposons un outil statistique permettant de choisir le compromis optimal, résultant en une

reconstruction robuste et de bonne qualité visuelle.

Ensuite, nous nous intéressons à la visualisation des fonctions reconstruites. Les fonctions de radiance sont définies soit par des attributs attachés au maillage, soit dans une texture. Dans les deux cas, le nombre de fonctions est fini et il faut pouvoir interpoler entre sommets ou texels pour déterminer une visualisation continue. Classiquement, les fonctions de radiance sont évaluées par sommet ou texel puis les couleurs résultantes sont interpolées afin de couvrir la surface. Nous définissons comment interpoler les fonctions avant de les évaluer et montrons pourquoi cela contribue à la qualité visuelle du maillage rendu. Ceci étend la définition de la radiance de façon continue sur la surface.

Finalement, nous traitons la simplification des données à visualiser. Un maillage comprenant des fonctions de radiance réparties sur sa surface peut s'avérer volumineux en mémoire et l'évaluation des fonctions est coûteuse en temps de calcul. Afin d'adapter la complexité à l'utilisateur et à son matériel, nous proposons un outil de simplification. Ceci se fait par un algorithme de contractions d'arêtes guidé par une métrique prenant en compte conjointement la dégradation en termes de géométrie et de radiance. Ce critère exploite l'interpolation de fonctions de radiance définies dans la contribution de visualisation.

Nous concluons en replaçant nos contributions dans le contexte scientifique et en proposant des perspectives de poursuite de travail.

Une annexe présente un travail que nous avons établi en parallèle de cette thèse. Il s'agit d'une méthode de génération de textures à la volée (*i.e.* au gré des besoins lors de la visualisation) basé sur une texture fournie en exemple. Elle génère à la volée un pavage infini à partir de l'exemple fourni tout en évitant les artefacts de répétition. Afin de pouvoir générer une texture fine à l'aide d'une représentation compacte, un mécanisme de transfert de couleurs entre différents niveaux de détails – chacun représentés par une texture fournie – est intégré et permet de générer une navigation naturelle dans une texture multi-échelle.

Nous introduisons et présentons ici l'algorithme de fragmentation qui est au cœur de cette méthode. Son résultat (le découpage de l'image en fragments) est à la base des deux contributions majeures : la création de diversité par l'échange de contenu entre différentes parties de la texture et la cohérence entre couleurs des différentes échelles.

Pour conclure, nous formulons une mise en lien entre nos travaux sur les champs de lumière et sur la génération de textures. Nous envisageons notamment une extension de cette dernière contribution à des textures de plus haute dimension (directionnelles ou bidirectionnelles).

Sommaire

Remerciements	i
Résumé	v
Préface	1
1 Introduction	3
1 Contexte de la numérisation et ses applications	5
2 Objectif général : reconstruire la géométrie et la radiance	7
3 Procédé et étapes de la numérisation	10
4 Numérisation de la forme	11
4.1 Acquisition par radio- ou photométrie	12
4.2 Reconstruction surfacique	13
5 Numérisation de l'aspect	15
5.1 Recalage de photos	15
5.2 Rétro-projection de pixels	16
5.3 Construction de fonctions de radiance	17
6 Traitement du champ de lumière surfacique	18
6.1 Visualisation	18
6.2 Simplification	19
7 Plan du mémoire	19
8 Notations	20
I Contexte scientifique	21
2 Représenter le monde réel	23
1 Rendu basé image	25
1.1 Échantillonnage et réduction de dimension	26
1.2 Applications résultantes	29
1.3 Bilan : photo-réalisme et coût mémoire	32
2 Simulation de la réflectance surfacique	33
2.1 Fonctions de réflectance	33
2.2 Représentations analytiques de la réflectance	36
3 Enjeux et méthodes croisées	40
3 Champs de lumière	43
1 Champ de lumière plénoptique	45
1.1 Acquisition et représentation	46

1.2	Rendu	47
1.3	Limites	47
1.4	Variantes de paramétrage	49
1.5	Besoin de géométrie	50
2	Champ de lumière surfacique	51
2.1	Radiance surfacique	51
2.2	Représentations	52
2.3	Nos choix	53
3	Fonctions de radiance	54
3.1	Modèles de couleur	55
3.2	Modèles de fonction	61
4	Conclusion	67
II Travaux réalisés		69
4	Reconstruction robuste de fonctions de radiance	71
1	Ajustement de fonction	75
1.1	Approcher les pixels directionnels	76
1.2	Choix de la métrique	77
2	Robustesse de la reconstruction	80
2.1	Défaut de robustesse	80
2.2	Stabilisation de l'ajustement	83
2.3	Résultats et comparaison qualitative	86
2.4	Bilan de la stabilisation	94
3	Analyse statistique	96
3.1	Biais et variance	97
3.2	Compromis entre qualité et robustesse	98
3.3	Résultats quantitatifs	103
3.4	Solutions pratiques	104
4	Conclusion	106
4.1	Bilan	106
4.2	Perspectives	107
4.3	Ouverture aux textures polynomiales	108
5	Visualisation de fonctions de radiance sur la surface	111
1	Stockage et interprétation de la radiance	113
2	Rendu d'un maillage	116
2.1	Visualisation sur GPU	116
2.2	Rendu de la couleur simulée	118
2.3	Rendu de couleurs stockées	121
3	Rendu de radiance	122
3.1	Interpolation classique	122
3.2	Interpolation de radiance « symétrisée »	125
3.3	Implémentation	126
3.4	Résultats	130

4	Conclusion	133
4.1	Bilan	133
4.2	Perspective : radiance par texel	134
6	Simplification de maillage surfacique avec attribut de radiance	137
1	Simplification de maillages : une taxinomie	139
2	Algorithme de contraction d'arêtes	141
2.1	Critères géométriques	142
2.2	Critères couleur	145
2.3	Bilan de l'existant	147
3	Nouveau critère combinant géométrie et couleur	149
3.1	Critère sur la couleur	150
3.2	Calcul du gradient	152
3.3	Résultats qualitatifs	152
4	Critère combinant géométrie et radiance	154
4.1	Critère sur la radiance	155
4.2	Calcul de l'erreur	156
4.3	Résultats qualitatifs	156
5	Conclusion	159
5.1	Bilan	159
5.2	Perspectives	162
5.3	Discussion : contraction avec plongement optimal	163
III	Conclusion	165
7	Conclusion	167
1	Apports pour le rendu de champs de lumière	169
1.1	Rappels des objectifs	169
1.2	Nos contributions	170
2	Choix du modèle de radiance	173
3	Perspectives	175
3.1	Résumé des perspectives évoquées	175
3.2	Étude sur le choix des fonctions de radiance	176
3.3	Radiance par face	177
3.4	Reconstruction conjointe	179
3.5	Long terme	179
IV	Annexes	181
A	Algorithme de bootstrapping	183
B	Gradient sur un triangle	185
C	Génération à la volée de textures multi-échelles par l'exemple	187
1	Introduction & contributions	188

2	Les fragments	188
2.1	Mélange multi-échelle	189
2.2	Contenu inter-changeable	190
2.3	Algorithme de fragmentation	192
2.4	Propriétés importantes	193
3	Conclusion	194
3.1	Perspectives en lien avec nos travaux de thèse	194
	Listes des figures, tableaux et algorithmes	195
	Lexique	197
	Bibliographie	199

PRÉFACE

Apparence (Larousse) : « *Aspect, conforme ou non à la réalité, sous lequel quelque chose [...] apparaît à la vue ou à l'esprit* »

Cette définition, aussi vague soit-elle, suggère que l'apparence de quelque chose ne peut être décrite à l'aide de qualificatifs simples tels que « ceci est une table en bois » ou « cette balle est ronde et jaune ». L'apparence d'une table ou d'une balle, telle qu'elle apparaît à l'observateur, est bien plus complexe, incluant des effets induits par l'illumination, notamment ombrages et réflexions, intimement liés à la géométrie.

Nous pouvons raisonnablement distinguer une forme (*e.g. ronde, table*) et une couleur (*e.g. couleur bois, jaune*). La forme peut être décrite, littéralement ou mathématiquement, avec plus ou moins de précision, mais n'est pas un concept difficile à cerner. La couleur telle qu'elle apparaît à l'observateur est moins triviale à appréhender. Elle est le résultat de l'interaction entre l'objet, son environnement lumineux et le capteur avec lequel il est observé (*e.g. système visuel humain, capteur photo*).

En informatique graphique, plus précisément en modélisation et en rendu, la couleur apparente pour un observateur fixe résulte de la forme de l'objet et de la réflexion de l'environnement lumineux sur sa surface. En réalité, cette définition simpliste implique bien des paramètres cachés : la couleur d'un point de la surface d'un objet dépend de la position de l'observateur, de l'environnement lumineux et du matériau qui compose l'objet. L'environnement lumineux et le matériau peuvent être infiniment complexes, ce dernier pouvant réfléchir la lumière de différentes façons et être translucide, fluorescent ou phosphorescent.

Dans cette thèse, nous souhaitons numériser un objet dans son environnement d'origine. Ainsi, l'éclairage est fixé mais arbitraire. Nous représentons l'apparence d'un objet numérisé par une description géométrique détaillée de sa forme (plus précisément de sa surface) sur laquelle nous définissons des fonctions de radiance. Une telle fonction est attachée à un point de la surface de l'objet et détermine, en fonction de la position de l'observateur, la couleur de ce point.

Cette thèse s'articule autour de cette représentation. L'objectif est de visualiser un objet numérisé de façon à ce qu'il ressemble à l'objet réel. Pour ce faire, nous allons mesurer la forme de façon précise (par scans tridimensionnels) afin de reconstruire un maillage surfacique et d'y joindre des fonctions de radiance à sa surface. Nous nous intéressons à la construction de cette représentation à partir d'un échantillonnage photographique, au traitement permettant de réduire la complexité des données reconstruites et à en établir une visualisation haute qualité.

Chapitre 1

Introduction

Numérisation de la forme et de l'apparence des objets et son apport à la société.



RÉSUMÉ:

Ce chapitre introductif présente dans un premier temps le contexte économique et les enjeux de la numérisation d'objets culturels (sections 1 et 2).

Ensuite, nous décrivons les étapes du processus (section 3) nous permettant de mener à bien une numérisation. Il s'agit de la numérisation de la géométrie (section 4 puis de la photométrie (section 5) et enfin des traitements des données reconstruites (section 6). Nous y expliquons brièvement les étapes dont nous sommes utilisateurs (sections 4, 5.1 et 5.2) afin de nous mettre dans les meilleures dispositions pour pouvoir comprendre la suite du manuscrit et cerner les problématiques que nous traitons. Nous introduisons ensuite ces dernières dans les sections 5.3, 6.1 et 6.2.

Dans la section 7, nous présentons la suite du manuscrit et les notations utiles sont introduites dans la section 8.

Sommaire

1	Contexte de la numérisation et ses applications	5
2	Objectif général : reconstruire la géométrie et la radiance	7
3	Procédé et étapes de la numérisation	10
4	Numérisation de la forme	11
4.1	Acquisition par radio- ou photométrie	12
4.2	Reconstruction surfacique	13
5	Numérisation de l'aspect	15
5.1	Recalage de photos	15
5.2	Rétro-projection de pixels	16
5.3	Construction de fonctions de radiance	17
6	Traitement du champ de lumière surfacique	18
6.1	Visualisation	18
6.2	Simplification	19
7	Plan du mémoire	19
8	Notations	20

1 Contexte

Cette thèse se veut une contribution dans divers traitements liés à la numérisation, en particulier du patrimoine culturel. Il s'agit là d'objets tels que des statues, façades de bâtiments historiques, fresques, fouilles archéologiques, tableaux, etc. Nous nous intéressons dans ce cadre à la reconstruction d'un avatar virtuel d'un objet d'intérêt. Les doctorats réalisés à Strasbourg par MM. Frédéric LARUE [2008] et Lucas AMMANN [2010] ont également contribué à cet objectif. Le premier traite de la numérisation d'objets culturels en se focalisant sur la numérisation géométrique et en proposant des débuts de travaux sur la reconstruction de l'apparence. Le second traite de la visualisation temps-réel de tableaux en permettant notamment la mise en valeur du relief de la peinture.

Ces travaux et les nôtres sont en continuité : nous allons traiter des problèmes liés à la reconstruction de l'apparence d'objets culturels, ainsi qu'à la visualisation et la simplification des données reconstruites.

L'obtention d'une copie virtuelle bénéficie aux acteurs, industriels, scientifiques ou amateurs, portant un intérêt pour le patrimoine ou la culture. En effet, l'archivage d'objets culturels est utile à plusieurs escients :

Communication. Un modèle virtuel d'un objet (*e.g.* peinture, statue) ou d'une scène (*e.g.* fouille archéologique ou paléontologique) permet deux types de communication : entre professionnels et envers le grand public.

D'une part, de coûteux déplacements de biens ou de personnes peuvent être évités grâce aux copies numériques. Un expert ou chercheur dans un domaine pourra ainsi étudier un objet d'intérêt à distance.

L'utilisation du médium visuel permet d'autre part d'améliorer la communication envers le public non expert, dont la compréhension de la recherche et de son utilité forme un enjeu sociétal important. On peut ici penser à la mise en place de montages vidéo mettant en scène des objets réels virtualisés.

Mise à disposition universelle. Alors qu'une partie de la culture est à l'heure actuelle réservée aux élites ayant les moyens d'y accéder, sa virtualisation permet de garantir l'accès à faible coût pour tous. De nombreux musées virtuels voient actuellement le jour et permettent ainsi de visualiser une peinture ou un objet dont l'original est présent à l'autre bout du monde.

Archivage. Certains éléments du patrimoine peuvent être qualifiés de « condamnés ». Il s'agit du cas où l'humain va dégrader l'objet en question de manière volontaire ou non.

Un exemple de destruction volontaire se produit dans le cas de fouilles paléontologiques où l'extraction des morceaux retrouvés est indispensable pour découvrir le site complet. Les scientifiques du domaine sont ainsi demandeurs d'outils permettant la reconstruction virtuelle du site [Subsol *et al.*, 2011].

Un exemple de dégradation involontaire dû à la simple présence humaine est celui des peintures de la grotte de *Lascaux* découvertes en 1940. Le site a été



Figure 1.1 : Exemple d'application de la numérisation : restauration d'une statue détruite à partir de la numérisation des fragments. Images de [Arbace et al., 2013].

ouvert au public dès la fin de la seconde guerre mondiale. Il a dû fermer en 1963 du fait de dégradations liées au changement climatique local induit par le va-et-vient des personnes. Une reproduction de la grotte a été créée et permet conjointement l'observation illimitée et la préservation du site original.

Nombreux sont les exemples de la sorte : grottes, parchemins ou livres, statuettes peintes ou tombeaux. L'intérêt culturel serait limité si, au nom de la préservation, nous décidions de fermer à jamais ces sites ou d'enfermer ces objets. Produire une copie virtuelle en amont permet non seulement la visualisation par le grand public, mais peut également ouvrir des perspectives de restauration *a posteriori*.

Restauration. Grâce à l'émergence de nouvelles technologies telles que les imprimantes 3D, l'archivage du patrimoine permet de recréer des copies réelles à partir de données virtuelles.

Un bel exemple de restauration est celui de la statue *Madonna* du petit village *Pietranico* dans le centre-est italien (figure 1.1). Celle-ci a été fragmentée en plus d'une dizaine de morceaux après le tremblement de terre qui secoua la région en 2009. Chacun des fragments retrouvés a été numérisé et ré-assemblé virtuellement afin de déterminer les parties manquantes. Ces dernières ont alors pu être ré-imprimées afin de reconstituer la statue.

Ce projet fut également une réussite scientifique de par les publications disciplinaires en informatique graphique [Scopigno *et al.*, 2011] et en héritage culturel [Arbace *et al.*, 2013], ainsi que des communications grand-public [Callieri *et al.*, 2011].

La numérisation a ainsi de nombreuses applications dans le domaine spécialisé de la culture. Pour le grand public, l'utilisation de la numérisation 2D est déjà courante depuis des décennies. Le passage à la 3D est lent mais se fait petit à petit. On n'est à ce jour plus très loin de mettre en place des photocopieuses 3D commercialisables : il s'agirait de machines « tout en un » capables de numériser un objet et de le ré-imprimer en 3D.

Vocabulaire

Photométrie : mesure de l'aspect. Par abus de langage, nous désignons par ce terme l'aspect d'un objet de façon plus large. Nous parlerons ainsi d'acquisition de la photométrie (prise de photos), de descripteur photométrique de la surface d'un objet (*e.g.* couleur, radiance ou réflectance) ou de distance photométrique (distance dans l'espace correspondant à un descripteur).

Représenter la forme des objets 3D numérisés est incontournable pour toute application. Certaines requièrent également une représentation de l'aspect : par exemple l'archivage et la visualisation de peintures ou de fresques. Les textures forment une solution bien connue et étudiée. Leur utilisation en informatique graphique et en infographie est généralisée. Or, l'ajout de descripteurs photométriques complexes incluant des effets directionnels de la lumière (radiance, réflectance, etc.) augmente grandement le réalisme des modèles visualisés (voir figure 1.3). Ces descripteurs ne sont cependant toujours pas suffisamment mûrs pour une utilisation répandue. Les solutions présentées dans ce manuscrit permettent de faciliter l'utilisation des outils utilisés.

2 Objectif général

L'objectif principal est de reconstruire un avatar d'un objet réel tout en considérant les contraintes liées au cadre applicatif. Ceci implique des contraintes de *réalisme*, d'*automatisation*, de *compacité*, et de *rapidité* de visualisation, si possible à *coût raisonnable*. À ce jour, les seules copies virtuelles répondant à tous ces critères sont de simples photographies, autrement dit des projections 2D d'un espace restreint, à un instant donné et dans des conditions d'éclairage fixes. Elles forment la majorité des œuvres dans les musées virtuels.

Cependant, les efforts entrepris dans les communautés qui s'y intéressent tentent d'augmenter le réalisme, en passant à des données 3D et à un enrichissement des descripteurs photométriques. La création de scanners 3D permet ainsi de mesurer avec une précision raisonnable la forme d'un objet. La numérisation du *David de Michel-Ange* illustré en exergue de ce chapitre en est un exemple renommé [Levoy *et al.*, 2000]. La numérisation de la photométrie est quant à elle plus complexe : la couleur apparente d'un objet dépend de nombreux paramètres, dont ceux liés aux directions d'observation et d'éclairage ainsi qu'au matériau composant l'objet.

Force est de constater que les solutions satisfaisantes sont au prix de grands efforts matériels et/ou humains. La plupart des méthodes de numérisation de caractéristiques photométriques directionnelles se basent sur des acquisitions avec des structures rigides (voir figure 3.3 page 48 par exemple) ou une phase de reconstruction



Figure 1.2 : Numérisation d'un objet réel (gauche), résultant en une reconstruction fine de sa forme (centre) grâce à la représentation par maillage surfacique dense, composé de millions de sommets plongés dans l'espace (droite).

faiblement automatisée. Cette thèse se veut ainsi être une contribution dans le sens des efforts d'automatisation et d'accessibilité des méthodes.

Géométrie. Nous souhaitons obtenir une représentation détaillée de la géométrie de l'objet. Ainsi, nous faisons appel à un modèle de maillage surfacique finement maillé. Il s'agit d'une représentation de la surface composée de milliers voire millions de facettes. Dans notre cas, elles sont triangulaires tel qu'illustré dans la figure 1.2. La numérisation géométrique, autrement dit le passage du modèle réel au modèle maillé ne fait pas partie des étapes auxquelles nous contribuons scientifiquement : nous en sommes de simples utilisateurs. Pour des raisons de complétude et de curiosité scientifique, nous en donnons toutefois les principales étapes dans la section 4 de ce chapitre.

Photométrie. Il y a de nombreuses façons de représenter la photométrie d'un objet. Nous en faisons un tour d'horizon dans le chapitre 2. Dans la réalité, la couleur observée d'une surface est le résultat d'une interaction entre un environnement lumineux et des matériaux. Ces derniers réfléchissent la lumière de façon complexe : elle peut être absorbée, réfléchiée, déviée, etc.

Du fait de nos applications, nous traitons particulièrement des objets non déplaçables ou situés dans leur environnement de prédilection (une statue religieuse dans son édifice, une peinture attachée à un mur éclairé, etc.). Nous souhaitons *a minima* pouvoir représenter l'objet tel qu'il apparaît dans cet environnement. Étant donné l'environnement et le matériau de l'objet, la couleur peut être définie comme une fonction de la position et de la direction d'observation. Il s'agit du modèle connu sous le nom de *champ de lumière*. Ceci permet de représenter des caractéristiques directionnelles, tels que des reflets spéculaires (voir figure 1.3).



Figure 1.3 : *Champ de lumière de l'objet « double dragon » : on peut dire que l'objet est composé d'une forme, une couleur diffuse et des effets de couleurs directionnels. Ces derniers peuvent être remarqués lorsque le point de vue change (voir flèches blanches).*

Vocabulaire

De nombreux documents détaillent les quantités radiométriques et photométriques. Nous ne donnons ici que le vocabulaire utile à nos travaux. Pour une information plus complète, un glossaire circonstancié pour la numérisation peut être consulté, par exemple en annexe de [Larue, 2008].

Quantité radiométrique : quantité physique décrivant le flux énergétique émis ou reçu en un point.

Quantité photométrique : quantité physique décrivant le flux lumineux émis ou reçu en un point. Le flux lumineux est identique au flux énergétique pondéré par la sensibilité spectrale de l'œil humain. Il ne quantifie ainsi qu'un flux dans le domaine spectral visible.

Luminance énergétique sortante : flux énergétique émis par élément de surface dans une direction, exprimé en *Watt par mètre carré par stéradian*.

Luminance lumineuse sortante : flux lumineux émis par élément de surface dans une direction, exprimée en *lumen par mètre carré par stéradian*.

Radiance sortante. Selon le dictionnaire, le mot « radiance » est un néologisme issu de l'anglais et désignant la luminance énergétique. Par abus de langage, ce mot est souvent utilisé en informatique graphique pour désigner la luminance lumineuse.

Représentation. Le modèle de champ de lumière peut être défini de plusieurs façons. Afin de combiner géométrie fine et champ de lumière, nous avons choisi une représentation dite *surfactive*. En chaque point de la surface, la couleur est déterminée par une fonction renvoyant une couleur pour toute direction d'observation, autrement dit une couleur émise en toute direction sortante de la surface. Une telle fonction définit la *luminance lumineuse* sortante en un point de la surface de l'objet. Par habitude dans la communauté, nous parlerons cependant de *radiance sortante*¹. Son domaine de définition est un hémisphère (représentant les directions d'observa-

1. Par la suite, nous omettrons de préciser l'adjectif « sortante ».

tion) orienté localement selon la normale à la surface). Nous verrons les définitions et les enjeux liés à une telle représentation dans le chapitre 3.

3 Procédé et étapes de la numérisation

Concrètement, nous souhaitons passer d'un objet réel à un objet virtuel décrit par un champ de lumière surfacique. Vu que nous traitons d'objets potentiellement fragiles voire non transportables, un certain nombre de contraintes d'acquisition s'imposent. Ainsi, nous devons nous restreindre aux acquisitions *in situ* en *espace contraint*. Ceci implique l'utilisation de matériel mobile et léger, pouvant être déployé facilement.

La figure 1.4 illustre les étapes allant de l'acquisition physique à la visualisation du modèle virtuel. Nous avons choisi ce découpage afin de bien différencier les parties dont nous sommes simples utilisateurs de nos apports. Certaines étapes cachent ainsi volontairement des procédés complexes.

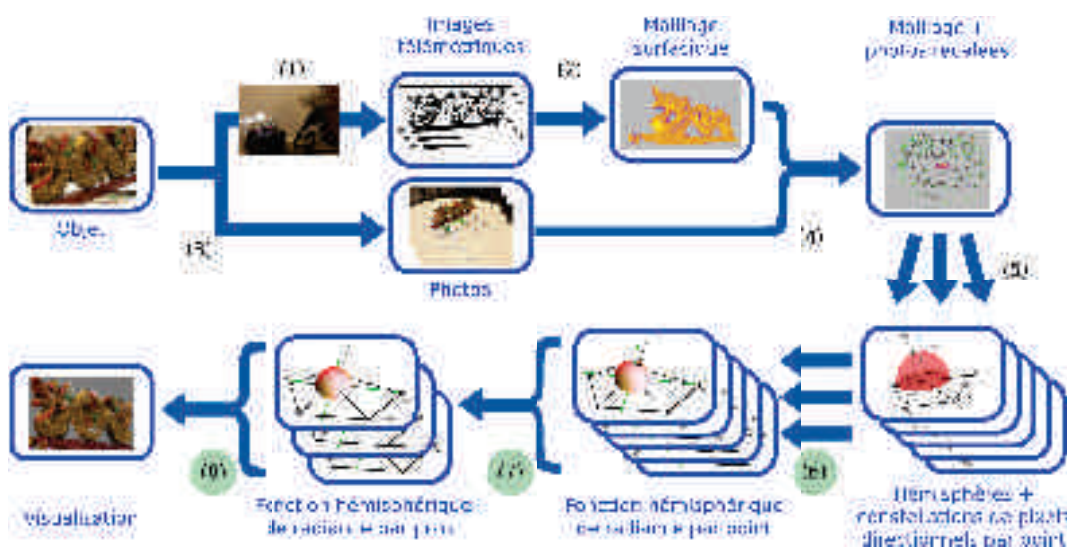


Figure 1.4 : Schéma décrivant le processus global de reconstruction d'un champ de lumière surfacique. Les numéros des étapes auxquelles nous contribuons sont apposés sur disque cyan.

1. L'*acquisition de la géométrie*, autrement dit la mesure par scanner 3D. Cette étape est détaillée dans la section 4.1 de ce chapitre.
2. La *reconstruction d'un maillage surfacique* comprend de nombreuses étapes algorithmiques permettant de passer de données télémétriques au maillage. Elles sont décrites dans la section 4.2.
3. L'*acquisition photographique* consiste en une simple prise de photos par un appareil photo que l'on peut trouver dans le commerce, et sous conditions d'illumination fixées et quelconques.

4. Le *recalage photographique* permet de situer toute photo dans le système de coordonnées du maillage. Ce procédé est décrit dans la section 5.1.
5. La *rétro-projection* de chaque pixel permet de déterminer le point de la surface duquel il provient. Autrement dit, il permet de connaître la couleur apparente d'un point pour différentes directions d'observation. La section 5.2 décrit ce procédé.
6. L'*ajustement de fonctions de radiance* traite localement et individuellement chaque point de la surface. Il permet de passer d'un ensemble de couples (couleur, direction d'émission/d'observation) à une fonction de radiance continue sur l'hémisphère. Le chapitre 4 traite cette problématique.
7. Une étape de *reformulation des données* peut contenir plusieurs choses, par exemple la compression dans un but de transmission ou de stockage ou encore le tatouage pour assurer une traçabilité. Nous nous concentrerons sur leur simplification (chapitre 6) permettant de déduire un modèle réduit avec un minimum de perte de qualité.
8. La *visualisation* permet de générer l'image résultante qui apparaîtra à l'écran pour un point de vue virtuel donné. Elle est discutée dans le chapitre 5.

Les étapes 1 à 4 sont décrites dans ce chapitre car elles ne font pas partie de nos contributions. Une connaissance basique de leur contenu est cependant utile, tant pour la complétude de ce manuscrit que pour la compréhension de certaines problématiques.

Les étapes 5 à 7 contiennent trois problèmes traités dans ce manuscrit. Les sections 5.3, 6.1 et 6.2 décrivent brièvement leurs problématiques et nous dédions un chapitre à chacune d'entre elles.

4 Numérisation de la forme

Cette section décrit le passage d'un objet réel au maillage surfacique reconstituant sa forme. Nous décrivons brièvement les étapes d'*acquisition physique* qui résulte en plusieurs images (photos ou images télémétriques) dont on pourra extraire un nuage de points 3D. La reconstruction permet ensuite de créer un maillage surfacique de bonne qualité. Ces étapes ne seront pas justifiées scientifiquement dans ce document. La plupart des informations décrites ci-dessous proviennent cependant de la thèse de doctorat de M. Frédéric LARUE [2008] où figurent davantage de détails. Il existe également quelques revues de l'état de l'art à ce sujet [Curless, 1999; Bernardini et Rushmeier, 2002; Botsch *et al.*, 2010].



(a) Acquisition par lumière structurée.

(b) Acquisition par temps de vol.

Figure 1.5 : *Scanners utilisés pour la numérisation au sein de l'équipe IGG, essentiellement composés d'un projecteur et d'un capteur visuel (caméra ou capteur laser).*

(1.5a) : projection de motifs lumineux structurés (sinusoïdaux) sur l'objet et déduction de la profondeur par vision. Ce scanner est utilisé pour mesurer un objet de petite taille et en intérieur.

(1.5b) : envoi d'un faisceau laser et chronométrage du temps de vol aller/retour vers l'objet puis déduction de la profondeur. Ce modèle est adapté aux grandes scènes : il est auto-rotatif et peut mesurer l'environnement autour de lui à 360° sur un rayon de quelques centaines de mètres.

4.1 Acquisition par radio- ou photométrie

Il s'agit dans un premier temps de faire les mesures de la forme de l'objet. Ces mesures se veulent non destructrices et basées sur du matériel portable.

Les approches récentes et désormais populaires se basent sur de simples appareils photos ou caméras pour extraire de la géométrie à partir d'images. Deux images prises de points de vue différents permettent de faire de la stéréo-vision et donc d'extraire de la profondeur (*multi-view stereo*) [Bao *et al.*, 2013]. L'acquisition d'un flot d'images de points de vue très proches permet de déterminer la géométrie avec des techniques de type « *structure from motion* » [Kim *et al.*, 2013].

Ces procédés permettent de générer un nuage de points 3D dans un repère unique.

Dans la plate-forme matérielle que nous utilisons², nous exploitons des scanners 3D qui sont des appareils permettant de mesurer la géométrie de façon détaillée. Autrefois chers et volumineux, ils sont devenus abordables et peuvent désormais être amenés sur site.

Les technologies les plus populaires sont :

- la mesure de la déformation d'un motif projeté sur l'objet. On parle alors de techniques à base de lumière structurée ou à décalage de phase [Salvi *et al.*, 2010].
- la déduction d'une profondeur par temps de vol aller-retour d'un signal lumineux envoyé sur l'objet.

Chacune de ces technologies a ses avantages et inconvénients en termes de précision,

². Voir : <http://icube-igg.unistra.fr/fr/index.php/Numérisation>.

de portée, de contexte d'utilisation, etc. Dans l'équipe IGG nous disposons d'appareils exploitant respectivement ces deux technologies (voir figure 1.5).

Le résultat d'une acquisition scanner est une image télémétrique, aussi appelée image $2\frac{1}{2}D$, représentant le champ de profondeur mesuré. Il s'agit donc d'une carte de distance entre la caméra du scanner et l'objet pour un certain nombre de directions (*i.e.* autant qu'il y a de pixels dans l'image télémétrique). Chaque image télémétrique peut être traduite en un ensemble de points 3D dans un repère propre à cette acquisition.

Ainsi, quelque soit le mode d'acquisition physique, on aboutit à un ou plusieurs ensembles de points 3D, éventuellement exprimés dans différents repères.

4.2 Reconstruction surfacique

La reconstruction de surface est un point très étudié en informatique graphique (voir l'article de Botsch *et al.* [2010]). Dans le cadre de notre plate-forme, nous reconstruisons un maillage de bonne qualité par le biais des traitement algorithmiques suivants :

1. *remaillage* des données partielles composant chaque acquisition,
2. *recalage* de ces données les unes par rapport aux autres,
3. *intégration* en un maillage unique, et
4. *débruitage* du maillage résultant.

1–Remaillage. Que l'acquisition ait été faite via des images télémétriques ou des photos, il est possible d'extraire une triangulation grossière à partir des nuages de points en exploitant la connectivité implicite fournie par l'alignement des pixels sur une grille. Pour la suite, il est possible de travailler sur trois représentations de données : image télémétrique, nuage de points ou maillage triangulé. Le choix dépend principalement des besoins de conservation des données.

Dans notre cas, nous déduisons immédiatement un maillage car la représentation sous-jacente de la surface permet de faciliter et d'améliorer la qualité du traitement qui va suivre, en particulier le recalage et l'intégration. Pour ce faire, nous avons utilisé l'algorithme de *Poisson* [Kazhdan *et al.*, 2006a], notamment parce qu'une implémentation est disponible [Kazhdan *et al.*, 2006b], ce qui en fait de plus un algorithme largement utilisé.

2–Recalage. Une mesure de scanner fournit l'information de profondeur à partir d'une unique position. Il est donc nécessaire de déplacer le scanner à plusieurs reprises (ou de déplacer l'objet mesuré, possibilité que nous omettons au vu de la nature des objets considérés) afin d'acquérir toutes les faces de l'objet. Il en résulte

plusieurs images télémétriques, donc plusieurs maillages partiels exprimés dans différents repères.

Deux manipulations préalablement effectuées lors de l'acquisition peuvent permettre de déduire la transformation rigide entre les données scannées :

1. Le marquage de la scène. Il s'agit de placer dans la scène des mires, visibles dans les scans. Elles permettent d'établir les relations entre données télémétriques.
2. La redondance de données. En s'assurant que chaque scan contienne de la redondance avec suffisamment d'autres, il est possible de déduire les transformations régissant les relations entre les différentes données.

Nous utilisons cette dernière méthode car la première est contraignante d'un point de vue physique : il faut pouvoir placer des mires dans la scène, voire sur les objets en question. Ceci n'est pas toujours possible dans le cas de données fragiles.

Le recalage se base donc sur le recouvrement des portions de surface reconstruites. Il peut se faire deux à deux de façon incrémentale ou globalement avec des algorithmes de minimisation itérative. Ces derniers étant lourds et sensibles aux minima locaux tout en étant plus précis si bien réglés, il est souvent d'usage de faire un premier recalage incrémental [Besl et McKay, 1992; Rusinkiewicz et Levoy, 2001] avant d'affiner les résultats avec une passe globale [Nishino et Ikeuchi, 2002].

3—Intégration. L'étape d'intégration opère la couture entre les différentes portions de maillages. Ces portions se recouvrent souvent un peu, tout en étant légèrement décalées les unes par rapport aux autres. Un recouvrement peut être obtenu en exploitant ces redondances et en ajustant une surface reliant chaque partie de façon lisse. Il suffit ensuite de faire une couture topologique pour fusionner les maillages.

4—Filtrage. Les données issues de scans sont inévitablement bruitées, et ce pour deux raisons :

- L'image télémétrique est discrète. Ainsi, le résultat ne peut être qu'approximatif. Un pixel est en effet le résultat de la projection d'un petit élément de surface – et non d'un point – de l'objet considéré. Dans cette petite zone, la surface n'est pas forcément plane ni orientée perpendiculairement à l'axe de projection. En ce sens, une approximation s'opère de fait.
- La précision numérique et physique des scanners induit également un bruit inéluctable. Il s'agit principalement de la limite en précision des algorithmes de calcul de la profondeur d'un point mesuré, mais d'autres paramètres plus complexes peuvent entrer en ligne de compte.

Ainsi, la rugosité du maillage reconstruit ne correspond que rarement à celui de l'objet original. Une surface d'objet réel que l'on qualifierait de lisse peut être reconstruite de façon rugueuse du fait d'un bruit géométrique haute fréquence. Un filtrage passe-bas adaptatif est souvent appliqué par souci d'esthétique. Il permet à

la fois de préserver les caractéristiques géométriques et de réduire le bruit [Mashiko *et al.*, 2004; Fleishman *et al.*, 2003].

5 Numérisation de l'aspect

Nous voulons déduire des fonctions de radiance à partir d'un ensemble de photos en entrée. Nous avons déjà évoqué qu'au vu des contraintes d'acquisition, nous nous limitons à un matériel léger.

Une solution classiquement utilisée, car nécessitant moins de travail algorithmique, est l'acquisition conjointe avec la géométrie [Larue et Dischler, 2006]. Ceci implique que chaque photographie soit associée à un scan (position identique ou proche).

La géométrie peut être numérisée en quelques scans (une à deux dizaines en règle générale). La durée d'un scan peut varier d'une à quelques minutes.

Inversement, il est souvent souhaitable d'acquérir la photométrie à partir d'un plus grand nombre de points de vue. Ceci est possible grâce à la portabilité du matériel et la rapidité d'une acquisition (moins de quelques secondes).

En somme, lier fortement les deux implique soit d'allonger considérablement le temps de numérisation, soit de diminuer l'échantillonnage photographique. Aucune de ces solutions n'est envisageable.

Ainsi, nous avons utilisé une procédure qui décorrèle les deux processus, gagnant ainsi du temps sans pour autant perdre en qualité. La prise des photos se fait avec un appareil numérique commercial haute définition³. Les emplacements des prises de vue sont déterminés manuellement. Empiriquement, on souhaitera tendre vers une couverture maximale et la meilleure répartition possible. Ceci peut être difficile du fait d'obstacles situés dans la scène ou de la grande taille de celle-ci.

5.1 Recalage de photos

D'après le modèle de Tsai [1992], le recalage des photos par rapport au modèle géométrique peut se faire en connaissant la *géométrie optique* de la prise de photo : *i.e.* les paramètres intrinsèques et extrinsèques du modèle projectif. Pour ce faire, nous avons utilisé un algorithme de type « *structure from motion* ». Il estime les paramètres de l'appareil en exploitant l'information mutuelle entre modèle géométrique rendu (avec un ensemble donné de paramètres virtuels) et image acquise [Corsini *et al.*, 2009].

Cet algorithme n'étant pas robuste, en particulier aux objets auto-similaires (tel que le double dragon de la figure 1.2 par exemple), nous avons ajouté des mires dans la scène photographiée. Ceci se fait de façon non intrusive et sans contact. Il s'agit des

3. Appareil *Canon EOS 5D Mk II* avec un capteur d'environ 20 millions de pixels.

cubes à damier dans la figure 1.2. Elles structurent fortement la scène et permettent de pallier aux problèmes de l’algorithme décrit ci-dessus.

Ceci nous fournit les paramètres de l’appareil photo mais pas encore de recalage des photos par rapport à la scène. L’algorithme *patch-based multi-view stereo* (PMVS) [Furukawa et Ponce, 2010] permet de déduire un modèle géométrique simple de l’objet à partir des photos⁴, et de connaître le placement des photos par rapport à ce dernier. Grâce à cela, un recalage avec le modèle géométrique détaillé acquis par radiométrie se fait facilement et le résultat est relativement précis. Un petit bruit sera cependant inévitable.

5.2 Rétro-projection de pixels

Suite à cela, il nous reste à déterminer le point surfacique d’origine de chaque pixel. En exploitant la géométrie optique désormais connue, nous pouvons procéder à l’opération de rétro-projection, qui est l’inverse de celle qui a constitué l’image. Ainsi, un rendu du maillage géométrique avec les paramètres d’appareil photo connus (position, modèle projectif) génère une image en correspondance directe avec la photo prise avec les mêmes paramètres. Une correspondance surface–image, et plus précisément portion de surface–pixel, peut alors être établie. Au final, tout point de la surface possède une constellation de ce que l’on appellera des *pixels directionnels* (voir figure 1.6).

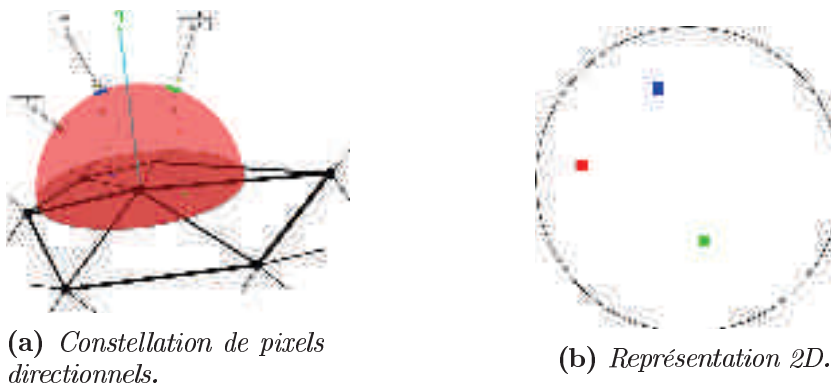


Figure 1.6 : Résultat de la rétro-projection de pixels : tout point de la surface possède une constellation de pixels directionnels (couleurs associées à une direction d’observation paramétrés sur l’hémisphère des directions). À gauche, visualisation 3D de l’hémisphère. À droite, visualisation 2D des mêmes données après projection orthogonale dans le plan tangent.

4. Le logiciel est disponible à l’adresse suivante : <http://www.di.ens.fr/pmvs/>.

Vocabulaire

Pixels directionnels : ensemble de couples (v, ω_r) associés à un point de la surface et issus de la rétro-projection de pixels acquis. Ces couples définissent la couleur v de ce point lorsqu'il est visualisé à partir de la direction ω_r . Si l'on utilise le vocabulaire du domaine de la « photo-simulation » plutôt que du « rendu basé image », on peut dire que la couleur c est émise dans la direction ω_r .

Le nombre d'éléments est variable entre les points et dépend de la couverture photographique et des occultations générées par l'environnement ou l'objet lui-même.

Notons toutefois qu'un pixel n'est pas originaire d'un point de la surface, mais d'une portion de celle-ci, dont l'étendue dépend notamment de la résolution photographique, de la distance entre le capteur CCD et la surface, de la distance focale ainsi que de l'orientation de la surface par rapport à l'axe de projection. L'utilisation d'une caméra haute définition (grand capteur CCD) permet de réduire ce problème, mais une certaine approximation est inévitable et difficile à quantifier.

5.3 Construction de fonctions de radiance

L'étape 6 du schéma 1.4 vise à construire le champ de lumière surfacique à partir d'un échantillonnage photographique. Notre contribution améliore ce procédé en termes d'*automatisation*, de *robustesse* et de *réalisme*.

De nombreuses solutions de reconstruction et de rendu de champs de lumière ne fonctionnent que lorsque l'échantillonnage photographique est très dense. Pour garantir cela, des dispositifs lourds permettant le positionnement d'appareils photos autour de l'objet (voire une variation de l'éclairage) ont été inventés [Debevec *et al.*, 2000; Hawkins *et al.*, 2001, 2005; Fuchs *et al.*, 2007b,a; Kohler *et al.*, 2013]. Notre contexte ne nous permet pas de tels dispositifs : numérisation *in situ* d'objets fragiles et/ou grands.

La prise manuelle de photos et le placement empirique de l'appareil photo génèrent un échantillonnage sans garanties de qualité en termes de répartition et de quantité. De plus, les diverses approximations évoquées (reconstruction géométrique, placement et rétro-projection des photos) introduisent du bruit. Lorsque l'on considère le problème localement par point de la surface, il apparaît qu'un point n'est visible que dans un sous-ensemble de photos. Cet ensemble peut être petit, du fait de la complexité géométrique de l'objet induisant des occultations. Ainsi, le nombre de pixels directionnels sera faible pour un point de la surface situé dans un creux par exemple.

Nous allons travailler sur l'ajustement de fonctions de radiance sur des pixels direc-

tionnels. Le résultat de cet ajustement est en règle générale fortement dépendant de l'échantillonnage local. Nous souhaitons au contraire que ce procédé soit robuste à de faibles variations.

À notre connaissance, il n'existe pas d'outil d'analyse de cette robustesse, ni de façon élégante de résoudre ce problème. La solution la plus communément utilisée consiste à sur-échantillonner, ce qui n'est pas souhaitable pour des données de champ de lumière. Nous allons faire le constat du problème de robustesse et suggérer une solution simple : stabiliser l'ajustement en minimisant une énergie. Ceci permet de définir un compromis entre robustesse d'une part et précision par rapport aux données mesurées de l'autre. Nous choisissons une énergie adéquate en analysant l'effet de plusieurs variantes, tant sur des critères de qualité visuelle comparée aux données d'entrée que de robustesse. Ceci se fait notamment à l'aide d'un outil statistique qui permet de définir le compromis optimal entre précision et robustesse.

6 Traitement du champ de lumière surfacique

Les étapes 7 et 8 du schéma 1.4 visent respectivement à simplifier les données reconstruites et à en établir une visualisation de qualité.

6.1 Visualisation

Notre contribution améliore le procédé de visualisation en termes de *réalisme*.

Les fonctions de radiance définissent la couleur en fonction de la direction d'observation en un ensemble discret de points de la surface. Cette direction est fixée lors du rendu d'une image : les couleurs peuvent ainsi être déterminées en évaluant ces fonctions. Il faut cependant pouvoir interpoler entre ces points afin de définir une couleur sur la surface entière.

Il est d'usage d'évaluer les fonctions puis d'interpoler linéairement les couleurs résultantes. Nous montrons que ceci est néfaste pour le rendu des caractéristiques directionnelles (en particulier reflets spéculaires). L'interpolation des fonctions avant évaluation est plus judicieuse mais deux verrous subsistent :

- Les fonctions peuvent être définies dans un repère de *Frenet*, donc localement. Ceci complique leur interpolation.
- Il faut tenir compte des reflets spéculaires dont les directions de réflexion dépendent des positions des sources lumineuses et de l'orientation locale de la surface.

La symétrisation des fonctions autour de la normale à la surface permet de tenir compte de ce dernier paramètre, ce qui va nous permettre d'expliquer comment définir une fonction interpolée souhaitée, tant mathématiquement que concrètement

sur la carte graphique. Nous illustrons qualitativement le gain obtenu.

6.2 Simplification

Notre contribution permet un gain en termes de *compacité* et de *rapidité de visualisation*.

Les fonctions de radiance sur le maillage forment un ensemble volumineux, ce qui peut engendrer des difficultés pour le stockage, la transmission ou la visualisation temps-réel. Une façon de résoudre ce problème est de définir une version simplifiée du même maillage, tout en prenant soin de le dégrader au minimum. Ceci implique de considérer la dégradation en termes de géométrie et de radiance de façon conjointe et cohérente.

Nous allons définir comment simplifier un modèle de champ de lumière numérisé à base de contractions d'arêtes. Nous proposons un nouveau critère permettant de mesurer une erreur locale commise à la contraction d'une arête et qui prend en compte la géométrie, la couleur et les effets directionnels inclus dans la radiance. Cette erreur se veut pertinente non seulement aux points où les fonctions de radiance sont définies, mais également en tout point de la surface où la radiance sera interpolée. Nous l'avons ainsi mise au point en cohérence avec l'interpolation définie dans le chapitre sur la visualisation.

7 Plan du mémoire

Nous souhaitons numériser un objet 3D en recréant tant la géométrie que la radiance. Pour ce faire, la géométrie est acquise par radiométrie et reconstruite avec divers algorithmes de reconstruction (étapes 1 et 2 de la figure 1.4). La photométrie est acquise par un ensemble de photographies dont les pixels sont rétro-projetés sur leurs points d'origine sur la surface (étapes 3 à 5 de la figure 1.4). Il en résulte un maillage surfacique triangulaire dense avec, sur sa surface, un nombre fini de points pour lesquels une constellation de couleurs est connue (voir figure 1.6).

À partir de cette étape débutent celles auxquelles nous contribuons :

- le passage d'une constellation discrète de couleurs à une fonction de radiance définissant la couleur de façon continue pour toute direction d'observation,
- la visualisation de données réparties discrètement sur le maillage de façon continue pour tout point de la surface,
- la simplification de ces données aboutissant à une version moins lourde tout en se souciant de minimiser la perte qualitative.

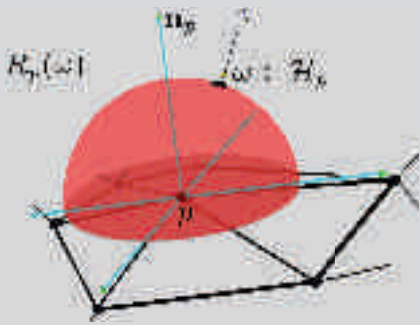
Ces contributions font respectivement l'objet des chapitres 4, 5 et 6. Elles ont pour point central la représentation par fonctions de radiance que nous utilisons.

Avant cela, nous plaçons cette représentation dans le cadre des travaux des domaines scientifiques en question. Nous faisons ainsi un tour d’horizon des méthodes de *rendu basé image* et de *photo-simulation* (chapitre 2). Nous décrivons ensuite les représentations de champs de lumière et définissons en détail les modèles que nous avons choisis de mettre en œuvre (chapitre 3).

8 Notations

Dans ce manuscrit, nous considérons des fonctions de radiance attachées à des points de la surface et aurons dorénavant recours aux notations suivantes.

Notations



p : un point de la surface auquel est rattaché une fonction de radiance. Il peut s’agir des sommets du maillage ou de texels d’une texture la recouvrant.

\mathbf{n} : une normale unitaire. La normale unitaire du point p est appelée \mathbf{n}_p .

\mathcal{H} : un hémisphère unitaire. L’hémisphère visible du point p est appelé \mathcal{H}_p et représente toutes les directions depuis lesquelles p est visible ou, de façon analogue, toutes les directions dans lesquelles p radie. Il est toujours défini tel que son pôle soit orienté dans la direction de la normale \mathbf{n}_p .

Ω : une sphère unitaire.

ω : une direction définie sur la sphère unitaire. Toute direction peut être décrite par deux coordonnées.

ω_i : une direction incidente, utilisée pour indiquer la direction de lumière reçue.

ω_r : une direction sortante, utilisée pour indiquer la direction de lumière émise (radiance) ou d’un point de vue.

R : une fonction de radiance sortante de \mathcal{H} dans \mathbb{R} associant un scalaire à une direction d’observation. La fonction de radiance sortante R_p associée à p est définie sur \mathcal{H}_p . Nous verrons que selon la représentation fonctionnelle choisie, elle peut être définie sur la sphère entière Ω plutôt que sur l’hémisphère.

PREMIÈRE PARTIE

CONTEXTE SCIENTIFIQUE

Chapitre **2**

Représenter le monde réel

« The body of the air is full of an infinite number of radiant pyramids caused by the objects located in it. These pyramids intersect and interweave without interfering with each other during the independent passage throughout the air in which they are infused ».

D'après Leonardo Da Vinci, ca. 1500, via [Adelson et Bergen, 1991].

RÉSUMÉ:

Nous présentons deux domaines historiques traitant des objets virtuels en informatique graphique. Le premier – le rendu basé image – s’occupe de reconstruire une réalité acquise par échantillonnage photographique (section 1). Le second – la modélisation ou la photo-simulation – part d’un modèle géométrique auquel sont joints des fonctions de réflectance de la surface et un éclairage virtuel (section 2). Tous ces éléments virtuels modélisés permettent alors de simuler la couleur résultante.

Nous présentons ici les enjeux de ces deux domaines ainsi que quelques représentations et techniques développées pour respectivement acquérir et rendre ou simuler avec réalisme. Dans la section 3, nous résumons brièvement les enjeux et expliquons que l’acquisition et le rendu d’un champ de lumière surfacique tel que nous le mettons en œuvre (*i.e.* le processus de numérisation vu au chapitre 1) exploite ces deux domaines : des données acquises par échantillonnage photographique sont transformées en des fonctions attachées à la surface.

Sommaire

1	Rendu basé image	25
1.1	Échantillonnage et réduction de dimension	26
1.1.1	Hypothèses sur la lumière	27
1.1.2	Restrictions d’espace et de temps	28
1.2	Applications résultantes	29
1.3	Bilan : photo-réalisme et coût mémoire	32
2	Simulation de la réflectance surfacique	33
2.1	Fonctions de réflectance	33
2.2	Représentations analytiques de la réflectance	36
2.2.1	Propriétés de la réflectance bidirectionnelle	36
2.2.2	Modèles courants de réflectance bidirectionnelle	37
2.2.3	Modèles de radiance sortante	39
3	Enjeux et méthodes croisées	40

Ce chapitre porte un intitulé bien présomptueux car il n'est à ce jour toujours pas possible de donner une totale illusion de réalité dans une scène virtuelle reconstruite. Nous allons cependant voir comment est simulé le monde réel afin de bien situer les approximations que nous faisons ou les contraintes que nous nous imposons lorsqu'on représente un objet avec un champ de lumière surfacique.

Notre méthode de numérisation et de reconstruction du champ de lumière se situe à la croisée de deux approches : l'une provenant du domaine du *rendu basé image*, l'autre de celui de la *photo-simulation* de l'illumination et de la réflectance. Leur fondement est historique. Le premier provient du domaine de la vision, davantage lié au traitement du signal, le second de la modélisation, plutôt liée à l'informatique graphique. Ces communautés ont été amenées à se côtoyer dans les années 1990 au rythme des avancées technologiques, et ont désormais une intersection d'applications conséquente.

Dans le *rendu basé image*, il est d'usage de caractériser une scène observée par des rayons de lumière traversant l'espace. Un rayon de la scène, observé à partir d'un point dans une direction et à un temps donné, en est un échantillon.

L'approche *photo-simulation* caractérise la réflectance lumineuse de chaque objet dans la scène. Ceci permet de déterminer la (quantité de) lumière réfléchie dans la direction d'observation (*i.e.* la radiance) étant donnés une certaine illumination et un point sur la surface.

Le procédé de numérisation que nous utilisons implique les deux approches : basé sur de la vision (photographies formant un échantillon des rayons lumineux traversant la scène), on déduit une fonction exprimant la radiance sortante. Après avoir présenté ces deux approches (sections 1 et 2), nous expliquons que ce procédé à l'interface de deux domaines permet de tirer l'avantage de chacune d'entre elles (section 3) : respectivement le photo-réalisme et le rendu temps-réel.

1 Rendu basé image

Ce domaine se base sur la fonction plénoptique¹. Citons l'article d'*Adelson* et *Bergen* [1991] introduisant cette fonction. Une traduction approximative de leur principe serait la suivante : « *Le monde est constitué d'objets tridimensionnels ne communiquant pas leurs propriétés à l'observateur. Elles remplissent plutôt l'espace autour d'eux avec des rayons lumineux qui constituent la fonction plénoptique. Une observation de cet espace constitue un échantillon de cette fonction. Le seul lien entre l'objet physique et les images rétiniennes est donné par la fonction plénoptique. Elle forme l'intermédiaire entre l'œil et le monde.* ».

Cette constatation est assez proche de celle de Leonardo Da Vinci citée en exergue, quoique plus formalisée. Elle a servi de base pour les travaux de numérisation de ce

1. « Plénoptique » : du latin *plenus*, pour complet ou plein, et *optique*.

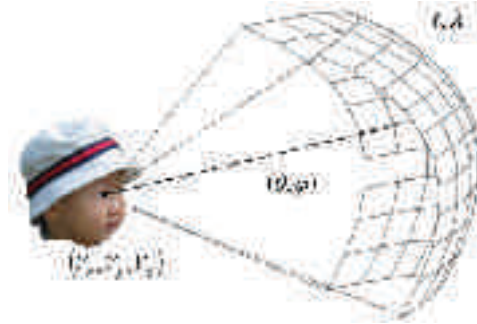


Figure 2.1 : Illustration de la fonction plénoptique déterminant la couleur observée en fonction de 7 variables. Image issue de [Zhang et Chen, 2004].

flot lumineux.

La fonction plénoptique peut être illustrée par la figure 2.1. Elle décrit un environnement dynamique 3D comme une fonction à sept dimensions $(V_x, V_y, V_z, \theta, \phi, \lambda, t)$. Cette fonction décrit le flux lumineux par point de l'espace $(V_x, V_y, V_z) \in \mathbb{R}^3$, par direction d'observation $(\theta, \phi) \in [0, \pi] \times [0, 2\pi]$, pour toute longueur d'onde λ et à tout temps t . Il est à noter qu'aucune géométrie n'est explicitement représentée.

Le rendu basé image a généré une très grande quantité de contributions scientifiques que nous ne pouvons toutes décrire dans cette thèse. Des articles passant en revue l'état de l'art du domaine peuvent être consultés pour plus d'information à ce sujet [Shum et Kang, 2000; Shum *et al.*, 2003; Zhang et Chen, 2004; Shum *et al.*, 2006].

1.1 Échantillonnage et réduction de dimension

Le *rendu basé image* est un procédé composé de deux étapes : l'échantillonnage du signal continu (*i.e.* la fonction plénoptique) et son rendu à partir des échantillons obtenus. Afin d'obtenir un rendu d'une certaine qualité, la théorie de l'échantillonnage (notamment le théorème de *Nyquist-Shannon*) peut donner une idée de la fréquence d'échantillonnage nécessaire. Il est cependant admis qu'échantillonner suffisamment la fonction plénoptique est irréaliste. Aucune tentative n'a d'ailleurs été rapportée.

Il est cependant possible de réduire la dimension de l'espace à échantillonner. Ceci se fait en appliquant des *hypothèses* ayant un coût en réalisme ou des *restrictions* ayant un coût de liberté de mouvement dans l'espace reconstruit.

1.1.1 Hypothèses sur la lumière

Hypothèse 1 : *Le flux lumineux de l'ensemble des longueurs d'ondes du spectre visible peut être approché par celui de trois longueurs d'ondes caractéristiques.*

Il est commun en informatique graphique de définir la couleur d'un rayon lumineux – autrement dit son flux lumineux pour toute longueur d'onde – par un triplet défini dans un espace de couleur tri-chromatique, par exemple l'espace RVB^2 . Il s'agit de sous-espaces de \mathbb{R}^3 permettant d'approcher une couleur.

Avantage : Plutôt que de déterminer – à temps, point et direction constants – un flux pour toute longueur d'onde afin d'acquérir la couleur par spectrométrie, il suffit de mesurer celui de trois longueurs d'ondes caractéristiques. Ainsi, *une triple acquisition* permet d'échantillonner l'espace des longueurs d'ondes, réduisant ainsi le problème d'une dimension.

De plus, cette approximation induit directement une *réduction de coût*. Les appareils permettant de mesurer un flux pour toute longueur d'onde (par exemple le spectromètre) sont en effet chers. Inversement, les appareils munis de capteurs tri-chromatiques (capteurs CCD dans la plupart des caméras et appareils photos) le sont peu et permettent par ailleurs d'acquérir les trois canaux en une passe. Le même raisonnement est valable pour les moniteurs d'affichage, que nous utilisons pour la visualisation.

Inconvénient : Cette hypothèse induit une faible perte de précision dans la mesure et la représentation des couleurs. Pour notre objectif de visualisation du modèle virtuel, l'approximation induite par l'hypothèse (1) est acceptable car presque imperceptible. Pour d'autres applications (*e.g.* mesures physiques de la couleur par spectrométrie), ceci ne serait pas le cas.

De plus, la visualisation se fait par l'intermédiaire d'un moniteur, dont les modèles commerciaux grand public exploitent la représentation de couleurs tri-chromatique RVB , considérant *de facto* l'hypothèse (1). Ainsi, il est courant en informatique graphique d'opérer cette approximation.

Hypothèse 2 : *Un rayon lumineux n'est pas altéré, ni en direction, ni en couleur, lors de son déplacement à travers un environnement constitué uniquement d'air.*

Avantage : Il n'est dès lors plus nécessaire d'échantillonner le même rayon à différentes distances de son point d'émission. La fonction plénoptique est alors définie par les rayons arrivants sur une surface arbitraire entourant la scène. *La position est ainsi réduite d'une dimension.*

Inconvénient : Exploiter cette hypothèse revient à utiliser la mesure d'un signal à un endroit et à le reconstruire à un autre. En soi, ceci n'est pas problématique,

2. Le chapitre 3 dédie une section à l'étude des espaces de couleurs.

mais une perte de qualité est inévitable du fait de la limite de résolution des capteurs numériques. En effet, un pixel d'une image couvre non pas un point mais une portion de la surface dont la taille est liée à sa distance au capteur (ainsi qu'à la distance focale et à la résolution du capteur). Ainsi, deux mesures le long d'un même rayon mais à des distances différentes encodent en réalité un rayon ayant une « épaisseur » qui diffère. Visualiser une telle donnée à une distance inappropriée induit des problèmes d'*aliasing*. Nous détaillerons ce point au chapitre 3. De plus, l'absence de représentation géométrique ne permet pas de connaître la distance de l'objet au capteur. Il est ainsi impossible de calculer la taille de la surface représentée par un pixel. Ceci est source d'artefacts de parallaxe.

Cette hypothèse est cependant communément utilisée dans la littérature [Gortler *et al.*, 1996; Levoy et Hanrahan, 1996]. Comme nous le verrons au chapitre 3 notamment, elle n'empêche pas l'obtention de résultats utilisables à condition de *sur-échantillonner* ou de *connaître la géométrie* de la scène.

1.1.2 Restrictions d'espace et de temps

La réduction de dimension peut aussi passer par la restriction de liberté de mouvement de l'espace reconstruit. Ceci peut être acceptable pour certaines applications.

Restriction 1 : *La scène est statique.*

Dans le cas où l'on s'intéresse à une scène statique – comme c'est le cas dans nos travaux –, *la dépendance au temps peut être éliminée* d'emblée, réduisant d'un la dimension.

Restriction 2 : *Le déplacement de l'utilisateur est limité à une surface prédéfinie.*

Lorsque l'on met en place une application de navigation dans un monde réaliste, il peut être raisonnable de restreindre le mouvement de l'utilisateur afin qu'il se maintienne sur un plan (le sol par exemple) plutôt que dans l'espace libre. Ceci réduit également la position de trois à deux dimensions.

Restriction 3 : *Le déplacement de l'utilisateur est limité à une trajectoire prédéfinie.*

Cette restriction réduit la position à une seule dimension.

Restriction 4 : *L'utilisateur ne peut pas se déplacer, il peut uniquement visualiser ce qui l'entoure.*

Si on fixe l'utilisateur à un endroit et qu'on ne lui autorise que de changer de direction de vue, on obtient une image (panoramique ou sphérique) autour de lui. Ceci élimine totalement la dépendance à la position.

Dans tout ce qui précède, nous ne nous abstrayons jamais des paramètres de direction d'observation (θ, ϕ) . Il y a deux raisons à cela.

La première est qu'il serait inopportun de qualifier une telle donnée de « basée image » au vu de la définition usuelle du mot image : « reproduction 2D d'une scène 3D projetée et visualisée à partir d'un point de vue fixe et pour un ensemble de directions d'observation définie par l'angle (solide) de vue ». Les dimensions (θ, ϕ) sont précisément les dimensions de l'image. Par ailleurs l'intérêt d'une telle représentation serait réduit en informatique graphique : visualiser qu'une seule direction (un pixel à l'écran) ou une courbe ne permet pas de représenter grand chose.

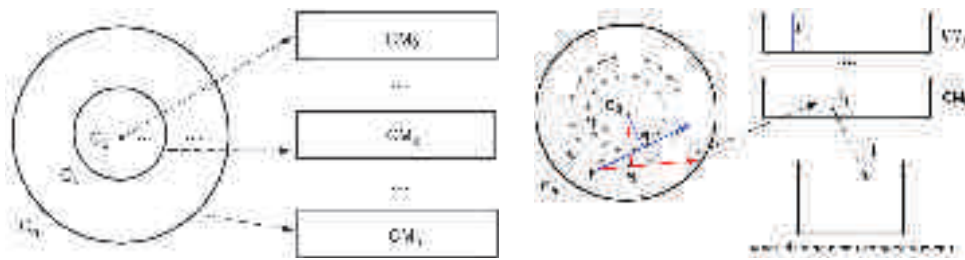
Deuxièmement, les capteurs utilisés (appareils photos et caméras) sont tous développés pour échantillonner un ensemble de directions d'observation en une seule action. Cet ensemble dépend de la géométrie optique du capteur (modèle projectif notamment).

Cependant, les appareils de mesure commerciaux usuels ne sont pas pour autant capables de mesurer tout l'espace 2D (la sphère englobante) en une seule fois. Des caméras omnidirectionnelles [Nalwa, 1996; Nayar, 1997] ou ayant des lentilles de type *fish-eye* [Xiong et Turkowski, 1997] peuvent s'en approcher. Comme nous le verrons ci-après, divers travaux traitent de la reconstruction de l'espace sphérique à partir d'un ensemble discret de photos l'échantillonnant.

1.2 Applications résultantes

Parmi les réductions de dimension citées ci-dessus, nos travaux considèrent les hypothèses (1) et (2) et la restriction (1) (scène statique). Ainsi, au prix d'approximations qualitatives, nous allons maintenir la liberté de mouvement de l'utilisateur (dans la limite d'une scène statique) et échantillonner puis reconstruire un signal à 4 dimensions (*i.e.* deux positionnelles et deux directionnelles). Vu que nous connaissons la géométrie, l'approximation induite par l'hypothèse (2) peut être corrigée par le calcul de la rétro-projection de pixels sur leur point de provenance.

Avant d'entrer dans les détails, nous proposons un tour d'horizon des diverses applications de reconstruction d'un sous-ensemble de la fonction plénoptique ayant vu le jour. La liste de travaux ci-dessous est représentative des applications mais non exhaustive. Plusieurs variantes et améliorations ne sont ainsi pas rapportées du fait de leur grand nombre.



(a) Acquisitions à partir de points dans l'espace situés respectivement en C_0 , sur le premier cercle (C_1 à C_k), sur le second (C_{k+1} à C_n), etc..

(b) Reconstruction d'une image pour une visualisation au point p . Les images acquises CM_k et CM_l sont notamment utilisées pour générer l'image.

Figure 2.2 : Mosaïques concentriques. Images de [Shum et He, 1999].

Nous répertorions ici les représentations et applications concrètes reconstruisant un signal allant de 2 à 6 dimensions.

2 dimensions. L'image est un échantillon à deux dimensions (angles de vue variables) de la fonction plénoptique après application des deux hypothèses et des restrictions (1) et (4). Une représentation 2D couvrant tous les angles peut être reconstruite à partir d'un ensemble d'acquisitions partielles. Ainsi, plusieurs travaux reconstruisent des *images panoramiques* sphériques ou cylindriques par procédé de *mosaicing*³ [Mann et Picard, 1994; Chen, 1995; McMillan et Bishop, 1995; Szeliski, 1996; Szeliski et Shum, 1997]. Pour éviter des effets de flou, les images doivent être acquises avec la même focale.

À la discrétisation des directions d'observation près, l'hypothèse (2) nous garantit la possibilité d'échantillonner complètement la scène visualisée. Une résolution suffisante permet alors de donner l'illusion d'une reconstruction continue et lisse dans l'espace des directions d'observation.

3 dimensions. La *vidéo panoramique* [Chen, 1995; Foote et Kimber, 2000; Neumann *et al.*, 2000] est une application à trois dimensions : la direction d'observation (2D) et la trajectoire (1D) de déplacement. Elle encode la fonction plénoptique après application des hypothèses (1) et (2) et des restrictions (1) et (3). Elle permet de se déplacer le long d'une trajectoire tout en observant son entourage en toute direction. Ceci est précisément ce qui est utilisé dans *Google street view* par exemple. Des applications de meilleure qualité existent également.

Une autre représentation 3D de la fonction plénoptique est la technique des *mosaïques concentriques* [Shum et He, 1999] (voir figure 2.2). Le résultat est équivalent à celui du panorama mais avec un degré de liberté de plus : cette représentation permet

3. « Assemblage d'une mosaïque » ici composé de plusieurs images.

d'ajuster la focale virtuelle lors du rendu. L'acquisition d'échantillons de la fonction plénoptique se fait le long de cercles concentriques autour d'un point, couvrant ainsi la position et la profondeur. Les rayons et la quantité de cercles concentriques doivent cependant être en proportion de la taille de la scène à représenter. Une scène extérieure n'est ainsi souvent pas échantillonnable de façon suffisamment dense.

On peut également mentionner la *vidéo classique*, qui est une représentation 3D de la fonction plénoptique après application des hypothèses (1) et (2) et de la restriction (4). Le temps et les deux directions d'observation (restreintes à celles captées en une acquisition par une caméra) sont en effet variables. La position peut également varier, mais n'est pas un degré de liberté car elle est déterminée de façon unique par le temps.

4 dimensions. À 4 dimensions, le *champ de lumière plénoptique* a suscité un grand intérêt de la communauté. Après prise en compte des hypothèses (1) et (2) et de la restriction (1), une description de la couleur de la scène à partir de tout point (2D) et en toute direction (2D) d'observation est obtenue. La problématique liée à ces représentations est le paramétrage efficace des rayons 4D. Les deux articles pionniers [Levoy et Hanrahan, 1996; Gortler *et al.*, 1996] considèrent un paramétrage par l'intersection de deux plans : le plan des positions et le plan focal. Les points d'intersection avec ces deux plans sont déterminés en fonction des paramètres de visualisation lors du rendu. Les rayons acquis les plus proches sont alors recherchés afin de définir la couleur du rayon virtuel.

5 dimensions. Une extension triviale en concept mais complexe à réaliser est le passage à une *vidéo de champ de lumière* [Yang *et al.*, 2002] (scène non statique). La quantité de caméras nécessaires pour acquérir cet espace est très grande : les tentatives de [Yang *et al.*, 2002] et [Wilburn *et al.*, 2002] en utilisent respectivement 128 et 64, et rapportent que ceci est néanmoins insuffisant pour éviter les problèmes d'*aliasing* et de *ghosting*.

Finalement, l'extension du champ de lumière en réfutant l'hypothèse (2) nécessite une description à trois variables pour la position [McMillan et Bishop, 1995]. Ceci résout le problème d'*aliasing* en échantillonnant également le long de l'axe de profondeur des rayons lumineux. Il est cependant complexe d'acquérir cet échantillonnage (les auteurs utilisent un capteur d'images cylindrique qu'ils déplacent dans la scène) et la quantité de données à gérer est également un obstacle.

6 dimensions. Au-delà de 6 dimensions, le temps de calcul est rédhibitoire. La *fonction plénoptique surfacique* reconstruit bien un signal 6D, en définissant la position sur une surface grâce à l'hypothèse (2). Or celle-ci n'a pas servi à élaborer une reconstruction utilisable, mais a uniquement été l'objet d'étude dans l'article

de Zhang et Chen [2003].

1.3 Bilan : photo-réalisme et coût mémoire

Le grand avantage de ces méthodes est de se baser sur des données par définition photo-réalistes. Le prix à payer est la gestion d'un grand volume de données et un risque d'erreurs liées à la distance focale.

Les reconstructions à deux et trois dimensions sont bien connues du grand public. La *light field camera* ou « caméra de champ de lumière »⁴ est désormais également construite et commercialisée depuis 2010 (par la société *Lytro* [2010]). Elle permet, au lieu de prendre une photographie d'un seul point de vue et dans une direction donnée, de photographier une scène à partir de différents points et directions de vues (très proches) en une seule acquisition. Le résultat est une image dont on peut modifier la focale et le point de vue a posteriori (sur un petit plan proche des points d'acquisition).

Une des raisons pour laquelle seul un petit plan d'acquisition peut être reconstruit est que l'échantillonnage de la fonction de plus de 3 dimensions génère des données trop volumineuses pour permettre une acquisition et un rendu rapide [Chai *et al.*, 2000; Zhang et Chen, 2003]. Les premières représentations de la fonction plénoptique (réduite) ne permettent ainsi pas un rendu en temps interactif : leur essor est passé par les nombreuses améliorations et simplifications qui y ont succédé, en particulier pour les champs de lumière. Ces améliorations du champ de lumière plénoptique traitent deux problèmes fondamentaux : le trop grand volume de données, rendant l'affichage en temps-réel difficile, et l'erreur de rétro-projection des pixels, générant des artefacts (*aliasing* et *ghosting*).

La plupart de ces méthodes se basent sur l'ajout d'informations externes, souvent de la géométrie (implicite ou explicite). Ceci permet d'établir des correspondances entre images, grâce à la connaissance de cartes de profondeur associées aux images ou de la géométrie complète de la scène. Ainsi, les erreurs peuvent être corrigées, ce qui permet d'éliminer le besoin de sur-échantillonner, faisant d'une pierre deux coups. C'est sans doute ceci qui a amené les communautés de rendu basé image et de modélisation (géométrique) à se côtoyer.

Nous n'établissons ici pas de liste des descripteurs géométriques qui ont rendu utilisables les applications mentionnées dans la section 1.2, notamment parce que cela n'a pas d'intérêt dans cette section (nous avons souhaité présenter les concepts et représentations issues de la fonction plénoptique), mais aussi parce que cela intersecte avec la section suivante et le chapitre 3.

4. Cette caméra a été définie en 1908 par *Gabriel Lippmann* [Lippmann, 1908] (prix nobel et directeur de thèse de *Marie Curie* notamment) dans un intéressant article historique dénommé « *Épreuves réversibles donnant la sensation du relief* ».

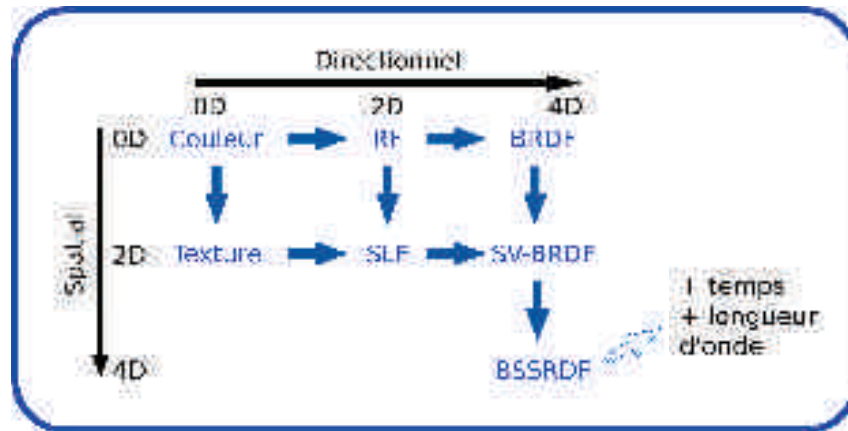


Figure 2.3 : Taxinomie des descripteurs de la couleur ou de la réflectance surfacique par dimension.

2 Simulation de la réflectance surfacique

Dans ce domaine, toute l'information de couleur est modélisée et ne fait donc pas recours à une quelconque numérisation. On ne caractérise cependant plus une scène complète en une fois mais un objet en particulier dont la géométrie est connue (précédemment modélisée ou acquise et reconstruite). La couleur rendue *in fine* sera calculée à l'aide de formules mathématiques appelées *fonctions de réflectance*.

2.1 Fonctions de réflectance

Les fonctions de réflectance simulent la réflectance d'une surface, *i.e.* la capacité à réfléchir la lumière. Les paragraphes qui suivent classent des descripteurs couramment utilisés pour décrire la couleur ou la réflectance d'un objet par ordre de dimension croissante. De fonctions de dimension nulles (couleur constante sur toute la surface), nous augmentons la dimension jusqu'à 12, permettant de décrire des phénomènes de réflectance complexes. Le schéma de la figure 2.3 illustre cette classification.

Couleur : une fonction constante. En langage courant, une information de couleur surfacique sans degrés de liberté reviendrait à décrire la couleur d'un objet par un qualificatif scalaire unique, par exemple : *cet objet est bleu*. La surface a ainsi une couleur sans dégradé ni reflets ou variations quelconques. Un tel objet est *a priori* inexistant dans la nature tant des paramètres extérieurs interviennent : la réflexion de la lumière, l'ombrage, etc.

RF : fonctions d'une direction. La notation RF (*radiance function*) du schéma 2.3 n'encode pas une fonction de radiance, mais la radiance sortante : elle définit la couleur en fonction de ω_r (2D). Une telle fonction se modélise difficilement. Elle est plutôt définie par ajustement sur des données de radiance acquises (voir SLF, décrit ci-dessous).

L'autre fonction 2D, dépendant de ω_i , s'appelle la *fonction d'illumination* et définit une couleur en fonction la lumière incidente. Elle peut servir à des fins de ré-éclairage : une source lumineuse virtuelle pourra être déplacée afin de définir la couleur observée à partir d'une direction d'observation fixe. Nous en verrons un exemple dans le chapitre 4.

BRDF : une fonction de deux directions. La fonction de réflectance bidirectionnelle (*Bidirectional reflectance distribution function* ou BRDF) a été définie dans les années 1960 par Nicodemus [1965; 1992]. Il s'agit d'une fonction à 4 dimensions $B(\omega_i, \omega_r)$ qui définit la proportion de flux réfléchi, et ce pour toute direction incidente ω_i et sortante ω_r .

$$f(\omega_i, \omega_r) = \frac{dL(\omega_r)}{dE(\omega_i)} \quad (2.1)$$

où L est la radiance (*i.e.* en réalité la luminance émise : cf. page 9) et E la radiance reçue. Autrement dit, il s'agit du flux émis vers un angle solide infinitésimal ω_r sur le flux reçu venant d'un autre angle solide infinitésimal ω_i .

La BRDF permet de caractériser la réflectance d'un matériau, ce qui donnerait en langage courant : « cet objet est un miroir » ou « cet objet est mat ». Il est ainsi courant en photo-simulation et en rendu d'en associer une au matériau d'un objet⁵ afin de pouvoir calculer la radiance dans la direction d'observation en fonction de la radiance entrante. Pour illustrer l'étendue de son utilisation, nous pouvons citer l'éditeur de BRDF créé à l'usage des graphistes des studios *Disney* [McAuley *et al.*, 2012; Walt Disney Animation Studios, 2012].

C'est l'équation du rendu [Kajiya, 1986; Immel *et al.*, 1986] qui calculera effectivement la couleur visualisée. Elle définit localement la radiance sortante en fonction de la BRDF et de la luminance incidente :

$$R(\omega_r) = \int_{\mathcal{H}} B(\omega_i, \omega_r) E(\omega_i) \langle \omega_i, \mathbf{n} \rangle d\omega_i \quad (2.2)$$

où B est la BRDF et $\langle \omega_i, \mathbf{n} \rangle$ un facteur d'atténuation prenant en compte que le flux lumineux est dilué sur une plus grande portion de surface pour une illumination rasante. Cette équation définit la radiance sortante d'un point d'une surface non émettrice de lumière⁶.

5. Par abus de langage, on désignera invariablement la réflectance d'un matériau ou sa BRDF. De même, la réflectance d'une surface ou d'un objet désignera celle du matériau le composant.

6. L'équation du rendu complète inclut d'autres paramètres : temps (pour une scène animée), et longueur d'onde que l'on remplacera par une représentation par espace de couleur (voir chapitre 3).

Hypothèse 3 : Additivité. *L'intégration des sources de lumière sur l'hémisphère des directions peut être approchée par une somme :*

$$\int_{\mathcal{H}} B(\omega_i, \omega_r) E(\omega_i) \langle \omega_i, \mathbf{n} \rangle d\omega_i \approx \frac{2\pi}{K} \sum_{k=1}^K B(\omega_i^k, \omega_r) E(\omega_i^k) \langle \omega_i^k, \mathbf{n} \rangle$$

L'hypothèse (3) suppose que la discrétisation de l'espace d'illumination n'altère pas ou peu la couleur visualisée. Ceci est vrai lorsque l'environnement lumineux est effectivement composé d'un nombre fini de sources lumineuses ponctuelles. Dans le cas contraire, l'erreur commise est petite si le nombre K de sources discrètes simulées est suffisamment grand et représente en priorité les sources d'éclairage principales. Lors du rendu d'une scène synthétique, cette hypothèse permet ainsi de simuler une carte d'environnement lumineux complexe en pré-définissant un nombre fini de sources lumineuses judicieusement placées. L'alternative plus coûteuse pour rendre une telle scène est d'opérer une intégration par une méthode de *Monte Carlo* [Metropolis et Ulam, 1949; Marques *et al.*, 2013].

Texture : une fonction de position. L'ajout de deux dimensions spatiales (voir figure 2.3) permet le passage d'une définition constante et globale de la couleur (*e.g.* « l'objet est bleu ») à une définition par position surfacique (*i.e.* « l'objet est dégradé du bleu au rouge avec un motif »). Concrètement, cela permet de définir une image 2D sur la surface de l'objet. Les éléments de cette image sont usuellement attachés aux sommets d'un maillage ou dans les texels d'une texture paramétrée sur l'objet. On parlera alors respectivement de *couleurs par sommet* ou de *texturage*.

La texture forme à ce jour la majorité des objets dont l'aspect a été numérisé. L'information colorimétrique permet un réalisme parfois acceptable, mais n'encode pas l'interaction avec la lumière.

SLF : une fonction d'une position et d'une direction. Les champs de lumière surfaciques (*Surface Light Field* ou SLF) sont le résultat d'une numérisation où l'environnement lumineux est arbitraire mais fixe. Il s'agit d'un ensemble de fonctions de radiance placées sur la surface afin d'en définir la couleur en fonction du point d'observation. Les champs de lumière surfaciques forment ainsi un espace 4D et les fonctions de radiance encodent le résultat de l'interaction de la réflectance de la surface et d'un environnement lumineux. Ceci peut inclure des phénomènes complexes tel que la réflexion de nombreuses sources lumineuses ou la réflectance sous-surfacique (ce qui n'est simulable qu'avec la BSSRDF, voir ci-dessous).

SV-BRDF : une fonction de position et de deux directions. La *Spatially-varying BRDF* (ou SV-BRDF) encode la BRDF locale, en tout point de la surface. Il s'agit d'une fonction à 6 dimensions qui permet de simuler le rendu d'un objet à surface opaque sous toute condition d'éclairage.

BSSRDF : une fonction de deux positions et directions. Certaines surfaces peuvent être translucides : un rayon de lumière peut entrer dans l'objet, rebondir de multiples fois en son sein, puis ressortir à un autre endroit. Ainsi, la *Bidirectional surface scattering reflectance distribution function* (ou BSSRDF) ajoute deux dimensions spatiales à la SV-BRDF, encodant l'endroit de sortie de la lumière [Jensen *et al.*, 2001]. Cette fonction a 8 dimensions.

Scènes complexes. Certains effets de réflectance ne peuvent être représentés par une BSSRDF : en particulier la *fluorescence* (*i.e.* émission d'un flux lumineux de longueur d'onde différente que celle reçue) et la *phosphorescence* (*i.e.* émission d'au moins une partie du flux lumineux au bout d'un temps non négligeable). Il faut ainsi ajouter deux variables caractérisant la longueur d'onde et une pour le délai de ré-émission, respectivement.

Représenter des scènes dynamiques simulant des surfaces qui changent d'apparence nécessite une variable temporelle en plus. On arrive ainsi à une description de la fonction à douze dimensions. Certains étendent cette définition à 14 dimensions, en passant à une représentation volumique : les couples de variables de localisation deviennent alors des triplets [Ramamoorthi et Biggs, 2009].

2.2 Représentations analytiques de la réflectance

La BRDF permet de représenter une large gamme de matériaux opaques. Nous analysons ses propriétés et quelques modèles mathématiques permettant d'encoder une BRDF afin de comprendre les comportements typiques en termes de réflexion lumineuse.

2.2.1 Propriétés de la réflectance bidirectionnelle

Les BRDF ont quelques propriétés intéressantes. Certaines sont toujours vérifiées, d'autres le sont pour certains types de matériau.

Propriété 1 : Conservation d'énergie.

Aucun matériau n'émet davantage d'énergie lumineuse (i.e. de flux lumineux) qu'il n'en reçoit.

Toute BRDF conserve l'énergie lumineuse : à direction d'incidence ω_i fixée, la valeur de la fonction intégrale sur l'hémisphère des directions sortantes ω_r multipliée par le facteur d'atténuation est inférieure ou égale à 1.

$$\forall \omega_i \in \mathcal{H}, \iint_{\omega_r \in \mathcal{H}} B(\omega_i, \omega_r) \langle \omega_i, \omega_r \rangle d\omega_r \leq 1 \quad (2.3)$$

Propriété 2 : Réciprocité de Helmholtz. *Un rayon lumineux, qu'il se déplace en un sens ou son opposé, encourt les mêmes phénomènes de réflexions, réfraction, atténuation, etc.*

Le théorème de réciprocité de la lumière, formalisé par Stokes [1883] sur base des travaux de von Helmholtz [1867] s'applique à toutes les BRDF. Il permet l'échange des angles de lumière incidente et sortante sans affecter la valeur de la BRDF [Hapke, 2005] : $B(\omega_i, \omega_r) = B(\omega_r, \omega_i)$.

Propriété 3 : Positivité. *La BRDF est non négative quels que soient les directions d'incidence et de radiance.*

$$\forall \omega_i, \omega_r \in \mathcal{H}, B(\omega_i, \omega_r) \geq 0 \quad (2.4)$$

Une BRDF non négative est dite *plausible*.

Propriété 4 : Isotropie. *La BRDF est constante lorsque l'on applique la même rotation autour de l'axe passant par le pôle de l'hémisphère à chacune des directions ω_i et ω_r .*

Un grand nombre de matériaux – mais pas tous – exhibent une BRDF dite *isotrope*. Dans ce cas, il suffit de trois variables au lieu de quatre pour encoder la BRDF.

Nous parlons dorénavant sans distinction de BRDF, de réflectance ou de fonction de réflectance d'un matériau. Lorsque nous évoquons la « forme » d'une BRDF, on désignera son comportement réflectif, illustré par la figure 2.4.

2.2.2 Modèles courants de réflectance bidirectionnelle

La forme d'une BRDF dépend du matériau sous-jacent. Ainsi, un miroir ne réfléchit pas la lumière de la même manière que de la terre par exemple. Certains modèles mathématiques décrivant une BRDF ont été utilisés en modélisation et en rendu. Ils approchent la réflectance des catégories de matériaux les plus courants par une représentation sous la forme d'une formule mathématique. En voici quelques exemples.

Modèle de Lambert. Une surface dite Lambertienne [1760] réfléchit le flux lumineux reçu de façon uniforme dans toutes les directions.

$$B_L(\omega_i, \omega_r) = \frac{\rho}{\pi} \quad (2.5)$$

où $\frac{\rho}{\pi}$ est une proportion (constante) de lumière réfléchi. Le résultat est donc indépendant du point de vue. Cette BRDF est en soi également indépendante de ω_i . De par le facteur d'atténuation, l'équation du rendu génère un résultat différent selon l'angle entre la normale en un point et ω_i . Ainsi, la quantité

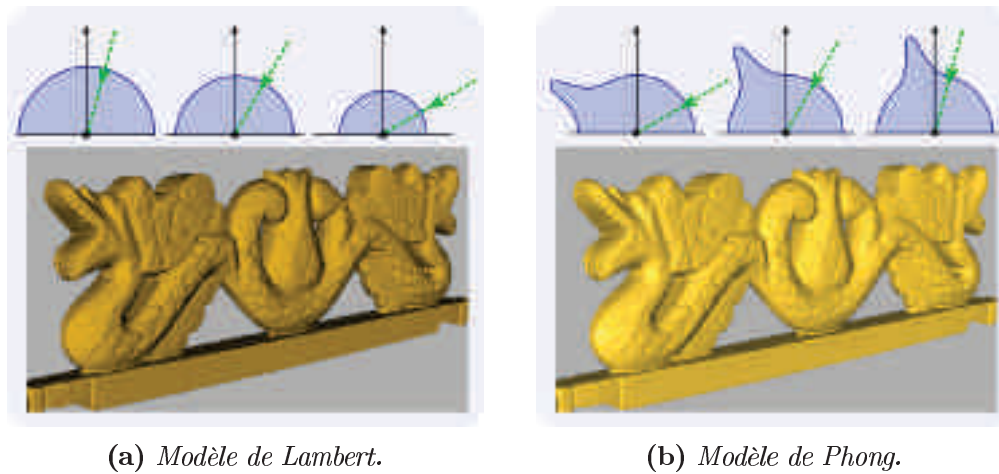


Figure 2.4 : Exemples de BRDF. En haut : représentations de la forme de la BRDF (isotrope). Les trois variantes représentent trois parmi l'infinité de directions d'incidence de la lumière (matérialisé par le trait vert). À angle d'incidence fixé, la courbe représente la radiance sortante : le flux lumineux émis pour toute direction d'observation. En bas : exemples de rendus avec ces deux modèles.

lumineuse réfléchi sera plus importante si le rayon de lumière arrive perpendiculairement sur la surface que s'il est rasant. Ce modèle permet de caractériser un objet mat ou purement diffus, tel qu'illustré dans la figure 2.4a.

Modèle de Blinn-Phong. Le modèle de Phong [1975] et Blinn [1977] permet de caractériser des objets brillants. Il définit le flux émis comme une somme pondérée d'une *couleur ambiante*, d'une *réflexion diffuse* (lambertienne) et d'un *lobe spéculaire*.

$$B_{BF}(\omega_i, \omega_r) = k_a I_a + k_d \langle \omega_i, \mathbf{n} \rangle I_d + k_s \langle \omega_r, \mathbf{r} \rangle^\alpha I_s \quad (2.6)$$

où \mathbf{r} est l'axe de réflexion idéal, obtenu par symétrie autour de la normale : $\mathbf{r} = \text{sym}_{\mathbf{n}}(\omega_i)$, où k_a , k_d et k_s sont des coefficients de pondération, où I_a est la couleur de la lumière ambiante, et où I_d et I_s sont respectivement les couleurs des composantes diffuses et spéculaires de la lumière incidente.

Définition

Lobe spéculaire : flux lumineux sortant réparti avec une intensité lumineuse plus forte autour d'une direction particulière et prenant la forme d'un lobe dans l'illustration de la figure 2.4a.

Il s'agit d'un modèle empirique non fondé sur une réalité physique : il s'applique hors cadre de l'équation du rendu et ne respecte pas la propriété (1) (conservation d'énergie). Dans ce modèle, le lobe est dirigé dans la direction de réflexion d'un miroir : l'axe symétrique au rayon lumineux en entrée par rapport à la normale de la surface. Ceci est illustré dans la figure 2.4b. Mathématiquement, la composante spéculaire est définie par un cosinus de l'angle

entre le vecteur de vue et celui de réflexion idéal, élevé à une puissance. La norme de cette puissance détermine la taille de l'angle solide dans lequel le lobe s'exprimera (*i.e.* sa largeur).

Autres modèles. D'autres modèles plus complexes ont été définis [Cook et Torrance, 1982; Beckmann et Spizzichino, 1987; He *et al.*, 1991; Nayar *et al.*, 1991; Torrance et Sparrow, 1992; Ward, 1992; Oren et Nayar, 1994; Ashikmin *et al.*, 2000]. Il s'agit soit de nouveaux modèles empiriques permettant d'améliorer le réalisme, soit de modèles basés sur la physique. Ces derniers permettent de prendre en compte la réflexion lumineuse générée par la micro-structure : une distribution statistique de micro-facettes est intégrée dans le modèle mathématique, ce qui permet de simuler correctement des matériaux rugueux. D'autres prennent en compte l'augmentation de la luminosité sur les directions rasantes ou la rétro-réflexion (qui consiste en un pic spéculaire dans la direction de la source lumineuse qui l'éclaire).

2.2.3 Modèles de radiance sortante

Alors que les fonctions de radiance ont deux degrés de liberté (une direction) de moins que la BRDF, elles doivent dans notre cas encoder des effets complexes contenus dans la radiance numérisée. Ceux-ci sont aussi arbitraires que l'environnement lumineux lors de l'acquisition photographique. Il n'existe ainsi pas de modèles de fonctions de radiance comme il peut en exister pour les BRDF.

Chaque fonction du haut de la figure 2.4 nous montre en réalité des radiances sortantes dans un cas simple : illumination par une seule lumière incidente fixe. Ainsi, selon la BRDF du matériau et l'environnement lumineux, des caractérisations de la radiance sont possibles : si l'objet numérisé a un matériau dont la réflectance se comporte comme une BRDF de type *Blinn-Phong* et que l'environnement lumineux est assimilable à une source ponctuelle, alors la radiance ressemblera à celle de la figure 2.4b : un pic spéculaire est dirigé dans une direction de réflexion donnée, qui aura été induit par la direction d'éclairage incident.

Dans le cas de l'acquisition sous conditions d'éclairage quelconques, les données acquises sont éclairées par un environnement lumineux complexe. Grâce à l'additivité (hypothèse (3)), la radiance sera alors formée d'une généralisation à plusieurs pics spéculaires orientés différemment et ayant des intensités lumineuses différentes. Un exemple schématique est donné dans la figure 5.7b de la page 126. Même si la radiance est difficilement « modélisable », elle peut néanmoins être représentée sous forme de fonction dans le cadre de l'acquisition de la radiance. Dans ce cas, des bases de fonctions « génériques » sont ajustées sur des échantillons acquis. C'est le procédé que nous mettons en œuvre dans nos travaux. Dans le chapitre 3, nous verrons différentes bases de fonctions qui peuvent être utilisées pour représenter une BRDF ou de la radiance, et qui sont bien adaptés pour l'estimation de paramètres à partir de données réelles.

3 Enjeux et méthodes croisées

Nous avons décrit le *rendu basé image* qui est un domaine proche de la vision et dans lequel *est née la numérisation colorimétrique*. Les difficultés y sont liés à :

- l'échantillonnage suffisamment dense,
- au traitement de la grande quantité de données qui en est le résultat, et
- au rendu interactif des données sur carte graphique.

Même si les capacités de stockage des cartes graphiques sont grandissantes, elles sont toujours limitées à quelques dizaines de textures⁷. En optimisant le stockage, il est au plus possible de stocker une centaine de photos, ce qui n'est pas suffisant pour bien caractériser l'apparence d'une scène visualisée dans le cas des champs de lumière ou de fonctions de dimension plus élevées. Il faut alors avoir recours à de complexes calculs hors ligne, comprenant de coûteux échanges entre carte graphique et processeur central.

À l'opposé, il y a l'approche *photo-simulation*, berceau de la *création de contenu par modélisation*. La difficulté y réside en

- la définition de modèles mathématiques pertinents, et
- la gestion de leurs paramètres.

Des modèles se limitant à des degrés de liberté directionnels (couleur et BRDF) ont été conçus dans des buts de modélisation : quelques paramètres permettent à des graphistes d'obtenir le résultat désiré. Des modèles avec variations spatiales (*e.g.* texture, SV-BRDF, BSSRDF) sont plus fastidieux à définir : il faut déterminer des paramètres pour chaque élément de surface. La définition d'une apparence photo-réaliste reste un problème d'actualité avec ces modèles (sauf peut-être pour la texture simple).

Méthodes à l'interface des deux communautés. Les deux communautés ont été amenés à se côtoyer afin de trouver une solution plus abordable à ces problèmes. Il est ainsi devenu populaire de créer du contenu par numérisation puis déduction de modèles mathématiques interpolant les données. Ceci cumule les avantages du « basé image » (*photo-réalisme*) et des modèles mathématiques (*compacité et rapidité*).

Beaucoup de travaux concernant la numérisation de la (SV-)BRDF ou de la BSSRDF (dont le concept est proche de la fonction de texture bidirectionnelle (BTF) [Dana *et al.*, 1999]) de matériaux réels ont ainsi vu le jour. Ces solutions sont complexes et matériellement coûteuses, tels le gonioréfectomètre [Foo, 1997; Li *et al.*, 2005] ou de grands dômes rigides [Isaksen *et al.*, 2000] permettant de positionner caméras et sources lumineuses à divers endroits sur l'hémisphère ou la sphère englobante afin d'obtenir un bon échantillonnage.

7. Les cartes graphiques récentes n'ont pas de nombre limité de textures, mais la taille totale est toujours limitée.

Il s'en suit une phase d'ajustement de modèles mathématiques sur ces données. Il y a deux tâches délicates : choisir le modèle adéquat et en estimer les paramètres en fonction des données. Ce sont ces problèmes qui vont nous intéresser au chapitre 4.

Finalement, il suffit d'évaluer les représentations mathématiques pour simuler la couleur rendue en fonction des paramètres virtuels (*e.g.* point de vue). La visualisation peut se faire efficacement dans les *shaders*⁸ de la carte graphique, exécutés de façon massivement parallèles.

Champs de lumière. Un problème plus abordable est la numérisation de la radiance (*i.e.* du champ de lumière). Les conditions d'éclairage sont fixées et n'ont ainsi pas besoin d'être estimées ou contrôlées, ce qui serait difficile hors laboratoire. La position de la caméra peut quant à elle être déterminée après les acquisitions (voir section 5.1 du chapitre 1, page 15) ce qui nous permet de placer la caméra manuellement, sans recours à des dispositifs lourds.

Nous nous intéressons à cette représentation pour deux raisons :

- le matériel léger nous permet de numériser des objets culturels, et
- représenter des champs de lumière surfaciques constitue une première solution satisfaisante dans notre cas : la numérisation de la radiance surfacique permet d'intégrer les effets dus à la réflexion induits par l'éclairage présent dans un environnement donné. En particulier, pour des œuvres fixées dans leur environnement « naturel » (*e.g.* fouilles, statues, peintures), ce sera leur éclairage habituel qui induira la radiance acquise puis reproduite sur la surface des objets.

Notre procédé se place à l'intersection des deux domaines. Les images acquises formant les données de base du champ de lumière plénoptique (sans géométrie) sont rétro-projetés sur un maillage acquis par ailleurs, afin d'ajuster des fonctions de radiance pour aboutir à un champ de lumière surfacique. Les modèles de champs de lumière seront étudiés plus en détail au chapitre 3.

8. Parties du *pipeline* graphique programmable. Nous en verrons l'architecture au chapitre 5.

Chapitre 3

Champs de lumière

« Champ de lumière » : défini par Arun GERSHUN [1939] comme « radiance » ou encore « une fonction de position et de direction dans l'espace non obstrué. Le champ de lumière est une fonction 4D, scalaire ou vectorielle selon la définition considérée. »



RÉSUMÉ:

Après avoir situé au chapitre précédent le champ de lumière dans le spectre des représentations de l'aspect d'une scène ou d'un objet, nous définissons plus exactement les modèles de champ de lumière. Ainsi, nous débutons par un tour d'horizon des représentations existantes (sections 1 et 2) avant de motiver notre choix de recourir au champ de lumière surfacique. Nous l'implémentons avec des fonctions de radiance attachées à la surface de l'objet et détaillons ici l'éventail de choix possibles pour mettre cela en œuvre (section 3). Parmi ces variantes, nous détaillons en particulier le choix de l'espace de couleur et d'une représentation analytique des fonctions.

Sommaire

1	Champ de lumière plénoptique	45
1.1	Acquisition et représentation	46
1.2	Rendu	47
1.3	Limites	47
1.4	Variantes de paramétrage	49
1.5	Besoin de géométrie	50
2	Champ de lumière surfacique	51
2.1	Radiance surfacique	51
2.2	Représentations	52
2.3	Nos choix	53
3	Fonctions de radiance	54
3.1	Modèles de couleur	55
3.1.1	Espaces de couleur	56
3.1.2	Comparatif	60
3.2	Modèles de fonction	61
3.2.1	Représentations analytiques	62
3.2.2	Comparatif	65
4	Conclusion	67

Dans ce chapitre, nous définissons le concept du champ de lumière plénoptique puis surfacique. De nombreux articles traitent des champs de lumière (plus de 1000 articles ayant le nom « *light field* » dans ses mots-clés parmi les publications de la *ACM digital library* uniquement), dont voici une sélection : [Sloan *et al.*, 1997; Isaksen *et al.*, 2000; Wood *et al.*, 2000; Chen *et al.*, 2002; Kitahara *et al.*, 2005; Levoy *et al.*, 2006; Cossairt *et al.*, 2008; Tompkin *et al.*, 2012; Kovacs et Zilly, 2012; Fuchs *et al.*, 2013]. Nombre de ces travaux sont des améliorations ou extensions par rapport aux concepts initiaux : ils permettent par exemple d'améliorer les performances de rendu ou encore d'augmenter le ratio qualité sur coût mémoire. Nous n'abordons pas ou peu ces améliorations mais présentons les concepts afin d'introduire les problèmes et de clarifier l'utilité du champ de lumière surfacique par rapport à son homologue plénoptique.

Pour rappel, nous cherchons dans cette thèse à déterminer puis à traiter et visualiser un champ de lumière surfacique à partir de données du champ de lumière plénoptique numérisé. Ce dernier décrit une scène visualisée depuis l'extérieur d'une surface englobante. Elle est 4D du fait qu'elle définit un signal 2D (image) pour tout endroit sur cette surface. Le champ de lumière surfacique détermine quant à lui une donnée de radiance en un ensemble de points de la surface d'un objet. La radiance en un point (2D) étant une fonction bi-variée (couleur dépendante du point de vue 2D), il s'agit ici aussi d'un signal 4D.

Si l'on restreint l'espace d'intérêt du champ de lumière plénoptique à un objet, ces définitions permettent de représenter les mêmes données. Dans le chapitre 1 (page 16), nous avons défini la rétro-projection de pixels plaçant les données d'entrée sur le maillage. Dans le chapitre 4, nous voyons comment ajuster une fonction continue sur ces données. Avant cela, nous faisons un état de l'art des représentations de champs de lumière, dans l'ordre chronologique d'apparition, avec un approfondissement de la représentation que nous avons choisie et adaptée à nos besoins.

1 Champ de lumière plénoptique

Le champ de lumière plénoptique est une restriction à 4 dimensions de la fonction plénoptique : par rapport à la fonction complète qui en a 7, il n'y a pas de dépendance à la longueur d'onde ni au temps. De plus, l'espace 5D qui en résulte est réduit d'une dimension en considérant que l'observation de rayons émis d'un même point et dans une même direction est équivalente en tout point le long de ce rayon. Le problème en rendu basé image est d'échantillonner cette fonction 4D pour stockage dans un premier temps. Ensuite, il s'agira d'exploiter les échantillons pour visualiser le champ de lumière, *i.e.* évaluer la fonction en tout point de son domaine 4D.

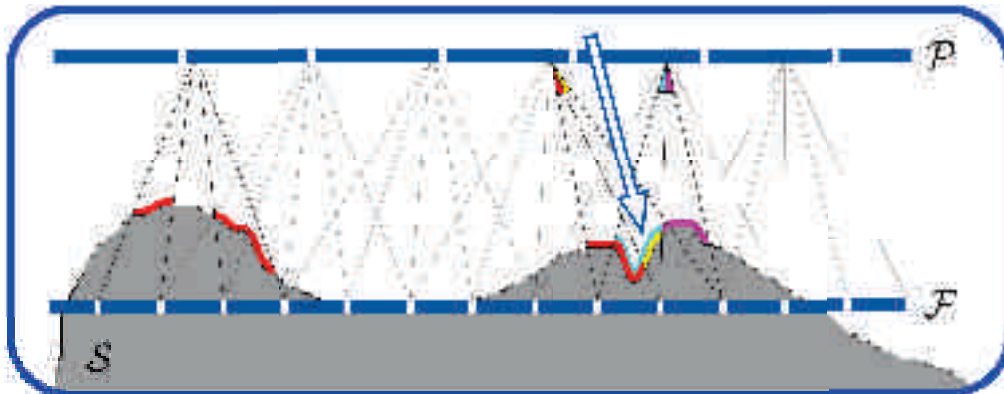


Figure 3.1 : Paramétrage selon deux plans parallèles (ici en 2D) du champ de lumière plénoptique. Tout rayon acquis est représenté par son intersection avec le plan des positions d'acquisition \mathcal{P} et le plan focal \mathcal{F} , représentant le point d'émission de la radiance (i.e. la surface « fictive »). La surface réelle \mathcal{S} peut s'en éloigner. À gauche : un rayon acquis en un pixel capture une portion de surface variable (surlignées en rouge) selon l'orientation et l'éloignement de \mathcal{S} . À droite : problèmes induits par la profondeur de champ. Le rendu du rayon bleu (flèche) interpole les rayons les plus proches (rouge, jaune, cyan et magenta), représentant des portions de surface très différentes.

1.1 Acquisition et représentation

L'acquisition échantillonne le champ de lumière plénoptique dont nous cherchons à obtenir un rendu continu. Pour ce faire, il convient de déterminer à quel rayon du champ de lumière plénoptique 4D continu correspond un pixel affiché à l'écran. Ceci est déterminé par le point de vue et la direction d'observation virtuels. Ensuite, ce rayon devra permettre de retrouver la couleur qui lui correspond parmi les données acquises.

Cet espace 4D a pour la première fois été modélisé en informatique graphique par Gortler *et al.* [1996] et Levoy et Hanrahan [1996] en parallèle : le *Lumigraph* et l'article *Light Field Rendering* proposent la même idée principale.

Il y est proposé de représenter tous les rayons 4D acquis comme intersectant deux plans parallèles : le plan des positions (2D) et le plan focal (2D). La figure 3.1 illustre cette représentation. Les acquisitions sont alignées sur un plan : le plan des positions \mathcal{P} . Le modèle sténopé utilisé couramment en photographie définit un plan sur lequel les acquisitions sont nettes : le plan focal \mathcal{F} . Tout rayon de la scène peut alors être exprimé par deux points : un sur chaque plan. Ceux-ci sont définis de façon continue, alors que les données acquises en forment un échantillonnage régulier.

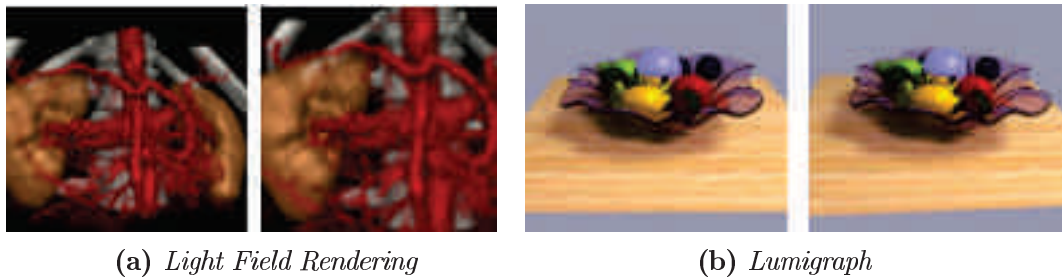


Figure 3.2 : Exemples de rendu de champ de lumière plénoptique. Les figures 3.2a et 3.2b sont respectivement issues de [Levoy et Hanrahan, 1996] et [Gortler et al., 1996].

1.2 Rendu

Le rendu est relativement trivial. Il consiste en deux étapes principales, effectuées par pixel à l'écran :

1. déterminer les deux points d'intersection du rayon avec \mathcal{P} et \mathcal{F} ,
2. interpoler les données les plus proches.

La première étape déduit les points d'intersection à partir des paramètres de la caméra virtuelle (voir flèche bleue dans la figure 3.1) : la position et la direction d'observation. La seconde étape opère par interpolation quadri-linéaire des 16 pixels – ceux correspondant aux rayons intersectant les 4 plus proches éléments sur chacun des plans – les plus proches de celui visualisé. La figure 3.2 montre deux exemples de rendu d'un champ de lumière plénoptique.

1.3 Limites

Même si l'idée de ces représentations est devenue populaire, de nombreuses limites ont rendu la version originale obsolète.

Acquisition rigide. Dans cette représentation, il est supposé que les caméras sont alignées sur une grille régulière correspondant au plan \mathcal{P} . Ceci permet d'interpoler les images de façon cohérente. Ainsi, il y a obligation d'utiliser un robot [Levoy et Hanrahan, 1996] ou des structures rigides, voire des pièces entières couvertes de caméras (figure 3.3). Ces solutions sont coûteuses et ne permettent pas de numériser tout objet, en particulier lorsque la numérisation a lieu sur site.

Erreurs de reconstruction. La configuration de la figure 3.1 (à gauche) suppose que la surface réelle \mathcal{S} (qui n'est pas acquise et donc inconnue lors de la reconstruction) est assimilable à un plan, *i.e.* le plan focal \mathcal{F} . Cette supposition permet



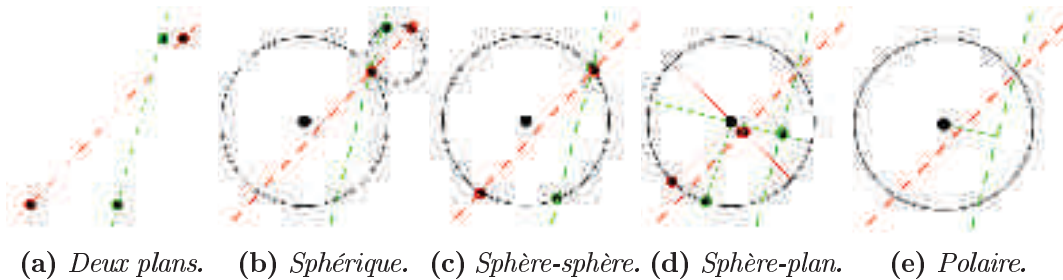
Figure 3.3 : Exemples de montage de centaines de caméras imaginés respectivement pour la numérisation d'objets culturels [Müller et al., 2005], pour des projets de réalité virtuelle [Vedula et al., 2005] ou de chirurgie/télé-conférence à base de réalité augmentée [Fuchs et al., 1994].

de paramétrer le plan \mathcal{F} en une grille régulière. Or, cette supposition est fautive en général. La surface est courbe et les pixels de l'image acquise ne correspondent pas aux éléments d'une grille régulière sur la surface, mais à une grille déformée qui est le résultat de la projection sur une surface non plane.

Ceci induit des *déformations* lors de la visualisation, en particulier à partir d'angles rasants la surface. L'interpolation entre pixels ne représentant pas des portions de surface de même taille amplifie cet effet. De plus, les effets de *parallaxe* ne peuvent pas être correctement pris en compte du fait de l'absence de connaissance liée à la géométrie. Il y a donc un risque certain qu'un point de vue s'éloignant de ceux qui ont été acquis génère une visualisation erronée. Finalement, la résolution du plan caméra étant finie, des effets de *flou* (aussi induits par les erreurs de parallaxe) peuvent apparaître lorsqu'on s'éloigne du plan \mathcal{P} . La figure 3.1 (droite) illustre ce phénomène : pour définir la couleur du rayon bleu, on interpolera les 4 (16 dans le cas tridimensionnel) rayons pointillés. Or, ces derniers contiennent des informations issues de portions de surface sensiblement différentes. Ceci génère un effet de *ghosting* (voir figure 3.2a par exemple).

L'erreur commise par l'hypothèse – l'écart entre \mathcal{S} et \mathcal{F} – est cependant à considérer en proportion inverse de l'écart entre les deux plans. Ainsi, elle peut être diminuée en éloignant les caméras (ce qui implique d'augmenter leur résolution pour maintenir une qualité constante). Néanmoins, une acquisition très dense est nécessaire pour assurer une visualisation correcte et l'utilisateur est souvent limité dans son mouvement dans l'axe perpendiculaire à la scène (translation avant/arrière).

Taille des données. Le sur-échantillonnage évoqué ci-dessus induit le problème de la taille des données. Des centaines d'images ne peuvent tenir en mémoire vive, et encore moins en mémoire vidéo. Les données doivent ainsi être compressées de façon efficace tout en permettant un accès aléatoire. Ceci est délicat, et souvent le gain en temps de transfert est au détriment de la qualité [Levoy et Hanrahan, 1996].



(a) Deux plans. (b) Sphérique. (c) Sphère-sphère. (d) Sphère-plan. (e) Polaire.

Figure 3.4 : Différents paramétrages des rayons du champ de lumière plénoptique. Images de [Schirmacher, 2003].

1.4 Variantes de paramétrage

Afin d’ajuster au mieux le modèle aux données, différents systèmes d’acquisition associés à de nouveaux paramétrages ont été développés. Ces dernières sont illustrées dans la figure 3.4.

Ainsi, certains ont préféré remplacer l’un et/ou l’autre plan par une sphère afin d’aboutir à un paramétrage angulaire uniforme. La sphère focale est ainsi plus adaptée lorsque la scène contient une géométrie davantage convexe et arrondie. De même, les caméras ont été positionnées régulièrement sur une sphère englobante au lieu d’un mur, afin de préserver les angles et ainsi d’éviter

- les discontinuités aux coins,
- la variation en distance caméra-centre de la scène, et
- les erreurs de projections obliques sur les caméras proches des bords du cube d’acquisition.

Un rayon a ainsi été décrit comme l’intersection

- d’un plan de positions et d’une sphère déterminant sa normale [Camahort *et al.*, 1998],
- de deux sphères [Ihm *et al.*, 1997],
- d’une grande sphère de positions et une petite sphère de directions judicieusement placée au point d’intersection avec la grande sphère, et représentant ainsi les directions d’observation locales à partir de ce point [Camahort *et al.*, 1998],
- d’une sphère et d’une coordonnée polaire [Todt *et al.*, 2008].

L’article [Camahort *et al.*, 2009] propose une étude des différents paramétrages, afin de les classer ou du moins d’en extraire les avantages et inconvénients. Cet article ne fait cependant que le constat irrémédiable que sans géométrie, un sur-échantillonnage est nécessaire, et que la déstructuration des acquisitions nécessite une phase de ré-échantillonnage régulier réduisant la qualité.

1.5 Besoin de géométrie

Suite au constat de Camahort *et al.*, la communauté s'est dirigée vers l'ajout d'autres données, en particulier de la géométrie. Ceci permet de rétro-projeter les pixels des images acquises sur la géométrie et ainsi de ré-échantillonner correctement les données afin de

- prendre en compte la parallaxe pour éviter la déformation des structures présentes dans l'image ainsi que le flou,
- diminuer la quantité de photos à acquérir,
- déstructurer l'acquisition.

Ajout de géométrie. L'article « *The Lumigraph* » [Gortler *et al.*, 1996] introduit déjà de la géométrie afin de corriger les erreurs de projection responsables d'un effet de flou (figure 3.2b). Pour ce faire, chaque rayon considéré pour l'interpolation est pénalisé selon plusieurs critères géométriques afin de ne donner du poids qu'aux plus appropriés.

Le « *view-dependent texture mapping* » (VDTM) de Debevec *et al.* [1996; 1998] utilise une géométrie simple – l'application visait des bâtiments – afin de pouvoir interpoler entre les images rétro-projetées. Les trois images de référence les plus proches du point de vue sont interpolées avec une pondération dépendante de la rétro-projection. Ceci est le premier algorithme temps-réel (notamment grâce à une faible quantité d'images).

L'article *unstructured lumigraph rendering* [Buehler *et al.*, 2001] fait un état des lieux de ce qu'un système de reconstruction de champ de lumière est sensé pouvoir accomplir (nous y reviendrons au moment du bilan dans le chapitre 7). Il définit un système qui répond à tous leurs critères. Il permet notamment d'exploiter de la géométrie lorsqu'il y en a, et de se réduire au *Lumigraph* lorsqu'il n'y en a pas ou peu. En particulier, il exploite une description géométrique pour permettre une liberté de placement des caméras lors de l'acquisition. Pour le rendu temps-réel, ces vues sont directement ré-utilisées (*i.e.* sans ré-échantillonnage) en interpolant un nombre arbitraire d'images (habituellement 4) choisies selon des critères bien précis, prenant en compte l'angle entre le point de vue et la surface, le ratio distance/résolution, etc.

Bilan. Malgré ces avancées, une qualité élevée sans artefacts nécessite une grande quantité d'images. L'effet de flou est réduit au minimum, mais n'a pas disparu : cela reste lié à la quantité, la densité et la résolution des images.

Nous allons voir dans la section suivante que ces problèmes peuvent être réglés avec une représentation compacte.

2 Champ de lumière surfacique

Le champ de lumière surfacique ne s'intéresse qu'à un objet unique dont la surface est supposée connue [Miller *et al.*, 1998; Wood *et al.*, 2000; Coombe *et al.*, 2005; Jachnik *et al.*, 2012]. Exploiter la géométrie permet de rétro-projeter les pixels et ainsi d'éviter les problèmes évoqués dans la section 1 (parallaxe, flou, quantité de données). Le champ de lumière surfacique redéfinit la façon de représenter les données. En effet, au lieu de considérer les rayons regroupés par point de réception (les positions et directions d'acquisition), elles sont ré-arrangées afin d'être regroupées par point d'émission (les positions de la surface et directions de radiance).

2.1 Radiance surfacique

Le champ de lumière surfacique est équivalent à la radiance surfacique. Il s'agit d'un espace à 4 dimensions : une couleur est définie par point de surface (2D) et par direction (2D). Dans une scène réelle, cette radiance est le résultat de réflexions sur la surface de rayons lumineux émanant de l'environnement. Ce sont les propriétés du matériau composant cette surface qui régissent la radiance générée par une illumination arbitraire donnée (voir chapitre 2). Le champ de lumière surfacique cherche à encoder cette radiance.

Avantages. La rétro-projection de pixels permet de se ramener à des données localisées sur la surface, ce qui implique un premier effet de bord intéressant : les données superflues issues des images acquises (les pixels ne se rétro-projetant pas sur l'objet d'intérêt) sont éliminées d'emblée. Ainsi, une partie conséquente des données est déjà écartée.

Ensuite, l'échantillonnage de l'espace représentant tous les rayons est creux : un point de la surface n'émet de la couleur que vers les directions extérieures à la surface, donc dans le demi-espace délimité par le plan tangent en ce point (voir figure 3.5). De plus, les auto-occultations (illustrées en partie droite de la figure 3.5) impliquent que certaines directions sont impossibles à acquérir.

Finalement, il s'agit d'un espace pouvant exhiber une grande cohérence : deux radiances voisines sur la surface, ou deux rayons proches émis du même point, ont souvent similaires. Ceci est en opposition avec deux images voisines ou deux pixels voisins dans une même image qui n'ont généralement que peu de choses en commun du fait des effets de lumière et d'ombrage ainsi que de la parallaxe.

2.2 Représentations

Nous séparons les représentations en deux catégories : les approches globales et les approches locales.

La représentation globale stocke les données brutes : un ensemble de rayons 4D dans $\Omega \times \mathbb{R}^2$. Le problème à résoudre est similaire au champ de lumière plénoptique : gérer la taille des données et déterminer les rayons à interpoler lors du rendu.

Rendu. Le rendu détermine les paramètres du rayon virtuel et trouve les rayons acquis les plus proches afin de les interpoler. La connaissance de la géométrie diminue la génération de flou et d'*aliasing* : on connaît précisément¹ la surface impactée par la rétro-projection d'un pixel. Ceci est donc pris en compte dans l'interpolation des rayons acquis. Si les artefacts persistent, cela sera du seul fait de l'interpolation qui a tendance à générer du flou.

Compression. La masse de données à gérer est inférieure à celle du champ de lumière plénoptique. Ceci n'exclut pas qu'elle puisse poser problème. De nombreux travaux traitent ainsi de la compression de ces données. Un accès aléatoire est toutefois nécessaire pour le rendu temps-réel. Ceci vient contraindre la compression.

Nous ne faisons pas de revue exhaustive des articles traitant de la compression. Voici une liste d'outils qui ont été appliqués aux champs de lumière surfaciques globaux :

- codage par blocs [Miller *et al.*, 1998] (à l'instar du codage JPEG),
- ré-échantillonnage basé sur le spectre de la transformée de Fourier [Chai *et al.*, 2000],
- quantification vectorielle (VQ) [Wood *et al.*, 2000; Lelescu et Bossen, 2004],
- transformée en cosinus discrète (DCT),
- transformée en ondelettes [Girod *et al.*, 2003],
- analyse en composantes principales (PCA) et décomposition en valeurs singulières (SVD) [Nishino *et al.*, 2001; Chen *et al.*, 2002; Coombe *et al.*, 2005].

Certains de ces outils sont naïfs alors que d'autres exploitent la cohérence évoquée. Ces méthodes de compression sont plus efficaces sur ces données que sur le champ de lumière plénoptique de ce fait.

La représentation locale décompose l'espace 4D en un ensemble de fonctions 2D directement localisées sur la surface [Wood *et al.*, 2000; Coombe et Lastra, 2006]. Ainsi une *fonction de radiance* est associée à chaque point de la surface, définissant la couleur en fonction de la direction à partir de laquelle elle est visualisée. Ces fonctions sont exprimées dans une base de l'hémisphère des directions extérieures par un ensemble de coefficients. Ceux-ci seront déduits par ajustement sur des données numérisées. Il s'agit de la représentation que nous utilisons dans nos travaux. Nous voyons dans la section 3 comment représenter et stocker ces fonctions.

1. À la finesse du maillage géométrique près.

Ajustement. Le chapitre 4 traite du passage local de pixels directionnels à une fonction de radiance. Pour le rendu, on dispose donc, en tout point p , d'une fonction continue $R_p(\omega)$ avec $\omega \in \mathcal{H}$ (ou Ω).

Rendu. Il s'agit de déterminer le rayon simulé pour tout p (*i.e.* la droite passant par la caméra virtuelle et p) et d'évaluer la fonction en la direction $\omega = \omega_r$ correspondante. Pour définir une couleur sur toute la surface, une interpolation surfacique de ces données sera également nécessaire. La contribution du chapitre 5 traite l'interpolation des fonctions sur la surface sur la surface.

Compression. Chaque fonction est définie par un ensemble de coefficients. Ainsi, la quantité de données à stocker est dépendante de deux éléments : l'échantillonnage spatial et la complexité mémoire de la représentation de fonction (*i.e.* le nombre de coefficients). En pratique, et afin de limiter le coût d'évaluation sur le matériel graphique, il est d'usage d'utiliser des représentations faiblement consommatrices en ressources, en particulier concernant la complexité unitaire d'une fonction. La complexité mémoire sera ainsi largement inférieure à l'union des pixels directionnels issus des données d'origine.

Notons que les fonctions peuvent être par la suite quantifiées en exploitant la cohérence entre fonctions de radiance car représentant un même matériau illuminé d'une même façon [Wood *et al.*, 2000]. Dans le chapitre 5, nous exploitons cette cohérence afin de maintenir une certaine qualité pour une résolution spatiale réduite.

2.3 Nos choix

Nous choisissons dans nos travaux d'utiliser une représentation locale du champ de lumière surfacique. Le passage du champ de lumière plénoptique au surfacique induit un gain sur tous les plans : le réalisme est augmenté par la correction de la rétro-projection (*i.e.* plus de parallaxe ou de flou) et la représentation est plus compacte. Cela répond de plus à notre besoin de représentation fine de la géométrie.

Les méthodes globales compressent les données brutes alors que les méthodes locales exploitent la compression résultante du passage aux fonctions décrites par un nombre fini de coefficients. Il est difficile de dire quelles sont les plus efficaces en ratio taux de compression sur qualité. Des études penchent vers les méthodes globales [Filip et Haindl, 2009].

Nous choisissons cependant d'utiliser les représentations locales pour le maintien du contrôle local et la flexibilité que cela induit en termes de traitement de maillage : les données géométriques et de radiance sont aisément manipulables de façon séparée. Ceci est primordial pour toutes nos contributions :

1. l'ajustement de fonctions se fait par un procédé local, répété en tout point d'attache p_i ,
2. la visualisation en tout point surfacique utilisera les données localisées en son voisinage, ce qui permet une implémentation GPU, et

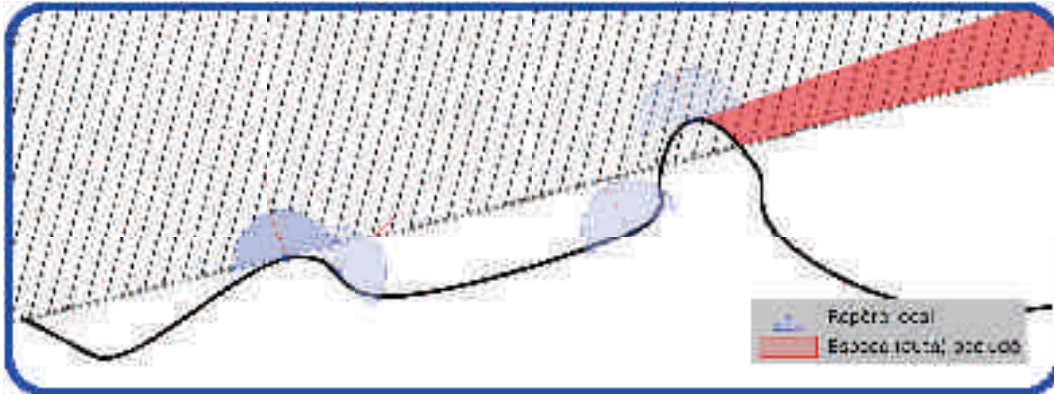


Figure 3.5 : Représentation du demi-espace à partir duquel un point de la surface est visible et orientations des hémisphères locaux.

3. la simplification exploite une métrique locale.

Avant de présenter ces contributions, nous faisons un tour d’horizon des façons de représenter et d’encoder les fonctions de radiance.

3 Fonctions de radiance

Les fonctions de radiance sont réparties sur la surface de l’objet. Chacune définit la couleur émise en leur point d’attache en toute direction. Elles serviront à définir la couleur de points du maillage en fonction de la position de l’observateur. En réalité, la radiance est définie en un point p par un triplet de fonctions, appelées « fonctions de radiance », dont chacune détermine la radiance d’un canal de l’espace de couleur E :

$$Rad_p : \mathcal{H}_p \rightarrow E : \begin{cases} R_1 : \mathcal{H}_p \rightarrow \mathbb{R} : \omega \mapsto C_1^T \Phi(\omega) \\ R_2 : \mathcal{H}_p \rightarrow \mathbb{R} : \omega \mapsto C_2^T \Phi(\omega) \\ R_3 : \mathcal{H}_p \rightarrow \mathbb{R} : \omega \mapsto C_3^T \Phi(\omega) \end{cases} \quad (3.1)$$

Dans ce cadre général, on peut faire divers choix : points d’attache au maillage, domaine, base et image de la fonction.

Points d’attache p . Les fonctions R peuvent être attachées de deux façons à une surface représentée par un maillage : à ses sommets ou à un texel déterminé par une coordonnée de la texture couvrant le maillage. Ce point n’est pas important pour notre première contribution (chapitre 4) qui traite chaque fonction individuellement. Nous détaillons donc les enjeux de ce choix au début du chapitre 5 où un récapitulatif de la structure de données générée par la reconstruction est présenté (section 1).

Domaine \mathcal{H}_p . Dans le cas d’une surface continue, chaque point est au plus visible à partir du demi-espace extérieur à l’objet – celui vers lequel pointe sa normale – délimité par son plan tangent (voir figure 3.5).

L'application de l'hypothèse stipulant qu'une couleur émise est constante en tout point d'une demi-droite de cet espace et passant par p permet de restreindre le domaine de définition spatial à l'ensemble des *directions* et non des points du demi-espace. Celui-ci sera dès lors représenté par l'hémisphère unitaire \mathcal{H}_p orienté localement avec la normale \mathbf{n}_p passant par son pôle. La direction $\omega_r \in \mathcal{H}_p$ étant à deux variables, R_p est une fonction bi-variée.

Nous verrons que certaines bases de fonctions prennent pour domaine la sphère entière Ω . Même si une bonne partie de ce domaine ne sera jamais utilisé lors de la visualisation, ceci a d'autres avantages que nous détaillons au début du chapitre 5 également.

Image E . R_p définit une couleur directionnelle. Son image est ainsi un espace de couleur tri-dimensionnel E . Plutôt que de définir une fonction vectorielle, nous définissons en p trois fonctions scalaires déterminant chacune la radiance d'un des canaux de E . En section 3.1, nous présentons et comparons les différents espaces de couleur.

Dans la suite, nous parlons de la fonction de radiance $R_p : \mathcal{H}_p \rightarrow \mathbb{R}$ d'un point. Il s'agit d'un abus de langage qui désigne en réalité les trois fonctions rattachées à ce point et qu'il est possible de faire parce que chaque canal est traité individuellement dans tout ce qui suit.

Base Φ . Une large gamme de représentations de fonctions est envisageable. Nous expliquons dans la section 3.2 que nous nous limitons aux bases $\Phi = [\phi_1 \dots \phi_N]^T$ afin d'exprimer la couleur comme une combinaison linéaire de coefficients $R : \mathcal{H} \rightarrow E : \omega \mapsto \sum_i c_i \phi_i(\omega)$, où $C = [c_1 \dots c_N]^T$ est le vecteur de coefficients déterminant les valeurs d'une fonction en particulier.

En section 3.2, nous présentons et comparons les différentes fonctions permettant de représenter la radiance.

Toute fonction rentrant dans ce cadre (*i.e.* celui de l'équation (3.1)) permet de représenter des fonctions de radiance. Il relève de l'utilisateur de faire un choix pour tous les paramètres évoqués d'après les critères qui lui paraissent importantes : qualité, compacité, rapidité, etc. Dans nos contributions, nous mettons en avant l'invariance de nos résultats par rapport à ces choix. C'est pourquoi nous en avons implémenté plusieurs. Les deux sections à venir présentent des variantes pour Φ et E , notamment toutes celles utilisées pour illustrer les chapitres à venir.

3.1 Modèles de couleur

Une couleur monochrome s'exprime par sa longueur d'onde. Or, une longueur d'onde ne permet pas à elle seule de définir toute couleur visible, souvent constituée d'une accumulation de plusieurs longueurs d'ondes monochromes. Ainsi, une description complète d'une couleur consisterait en sa décomposition en couleurs monochromes (*i.e.* sa décomposition spectrale). Ceci n'est pas réalisable numériquement, c'est pourquoi les couleurs sont approximées par des espaces à trois composantes.

L'inspiration première pour la composition d'un tel espace vient de l'œil humain. Ce dernier est muni de capteurs (les *cônes*) de trois types, chacun sensibles à une gamme de longueurs d'ondes différentes. Ceux-ci ont respectivement des pics de stimulation autour de longueurs d'ondes courtes (420 – 440nm, proche du bleu), moyennes (530 – 540nm, proche du vert) et longues (560 – 580nm, proche du rouge). Le cerveau interprète ainsi les couleurs par synthèse additive : il mélange les flux de couleurs rouges, vertes et bleues ressenties afin d'identifier la couleur qui en résulte. Plus le flux lumineux perçu est grand, plus la couleur sera perçue comme claire (d'où également la sensibilité voire la douleur oculaire lorsque l'on regarde une source lumineuse).

L'approximation faite par cette interprétation tri-*stimuli* est que l'œil perçoit une couleur monochrome (par exemple du jaune pur) et une couleur issue d'un mélange (par exemple du jaune issu d'un mélange de rouge et de vert) de la même manière malgré une composition physique en termes de longueurs d'ondes différentes. L'espace *RVB* (section 3.1.1) est une représentation numérique reprenant ces idées en faisant la même approximation.

3.1.1 Espaces de couleur

Trois fonctions de radiance vont permettre de déterminer trois composantes numériques d'un espace de couleur. Lors du rendu, chacune sera évaluée individuellement et un triplet de scalaires formera alors la couleur à afficher. Les espaces de couleur se différencient par leur structure mathématique : la façon de composer la couleur finale diffère. Les enjeux régissant le choix d'un espace de couleur sont les suivants :

Coût de conversion vers *RVB*. Par analogie avec la vision humaine, et parce qu'il y a un fondement physique basé sur la synthèse de couleurs, il est souvent d'usage en informatique d'utiliser l'espace *RVB* : la majorité des capteurs CCD, moniteurs, etc. utilisent cet espace. Choisir un autre espace exige donc de faire des conversions après l'acquisition ou lors du rendu. Ces conversions ont un coût dont il faut tenir compte.

Uniformité perceptuelle. La distance euclidienne entre deux couleurs n'est pas la même dans tous les espaces. Dans les chapitres à venir, nous aurons à faire des mesures d'écart ou des interpolations entre fonctions définissant une couleur. La proximité des distances mesurées avec celle perçue par l'humain est ainsi une propriété importante des espaces de couleur.

Définition

Espace perceptuellement uniforme : espace de couleur dans lequel la distance euclidienne est proportionnelle à la distance perçue par l'humain.

Plus l'espace utilisé se rapproche de l'uniformité perceptuelle, plus les calculs seront proches de ce qu'attend l'observateur.

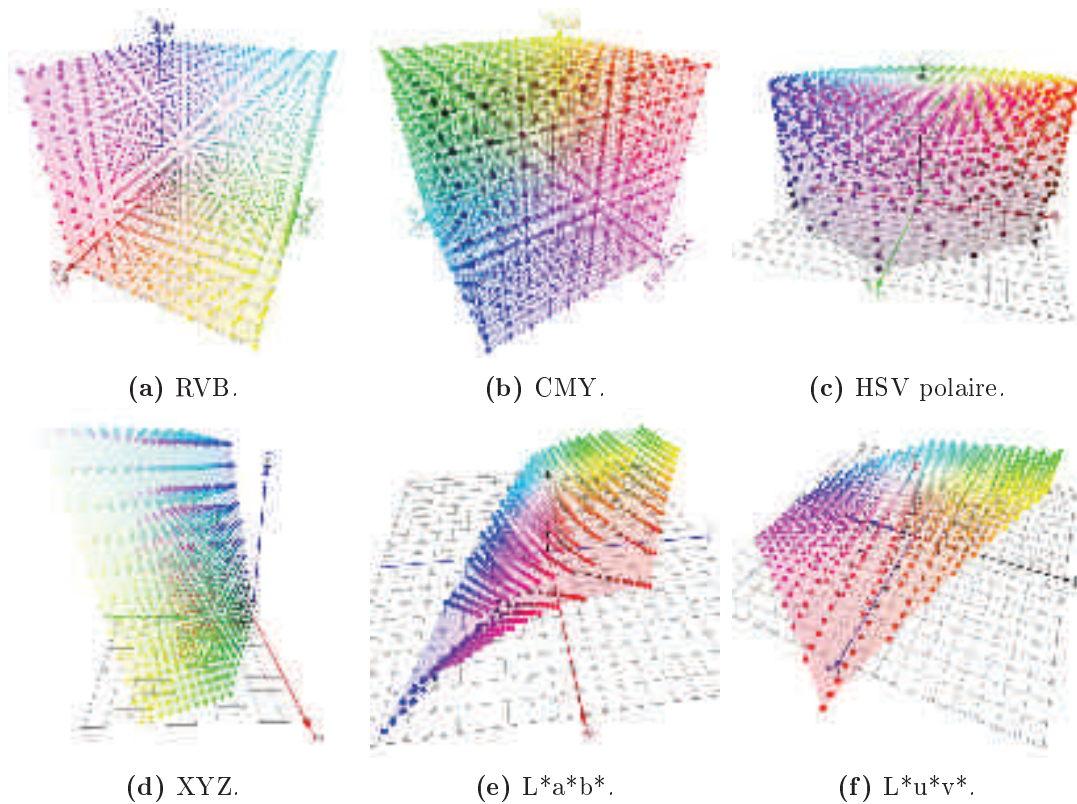


Figure 3.6 : Visualisation des espaces de couleur. Source : <http://www.couleur.org>.

Nous passons ici en revue différentes familles d'espaces de couleur et les comparons ensuite en fonction de ces critères. La figure 3.6 sert d'illustration à nos propos.

Espaces cubiques. Deux espaces sont basés sur le mélange de couleurs primaires : l'espace *rouge-vert-bleu* (RVB) et l'espace *cyan-magenta-jaune* (CMJ). Ils vivent tous les deux dans un sous-espace euclidien $[0, 1]^3 \in \mathbb{R}^3$ ayant pour base trois vecteurs orthogonaux directs². Chaque canal décrit la quantité respective de chacune des composantes à mélanger afin de composer une couleur. La conversion d'une couleur réelle (décrit par son spectre continu) vers un espace de couleur tri-chromatique se fait à l'aide de la fonction d'efficacité lumineuse spectrale [Commission internationale de l'éclairage (CIE), 1988].

L'espace *RVB* se base sur la synthèse additive : les quantités de rouge, de vert et de bleu s'ajoutent pour tendre vers le blanc, alors que trois valeurs nulles représentent le noir. Ceci peut être illustré sur un cube unité : les trois sommets de coordonnées $[0, 0, 1]$, $[0, 1, 0]$ et $[1, 0, 0]$ correspondant aux trois couleurs respectives (voir figure 3.6). La coordonnée $[0, 0, 0]$ représente le noir et la coordonnée $[1, 1, 1]$ le blanc.

2. Une variante souvent mise en œuvre est l'encodage sous forme d'entiers sur $[0, 255]^3$.

L'espace CMJ utilise la synthèse soustractive (utilisée en impression et en peinture notamment) : l'addition des composantes tend vers l'absence de couleur, donc le noir (coordonnée $[1, 1, 1]$) et la coordonnée $[0, 0, 0]$ représente inversement le blanc. Cet espace est également représentable sur le cube unité.

Il existe une bijection évidente entre les coordonnées RVB et CMJ d'une même couleur :

$$\begin{aligned} [r, v, b] &= [1 - c, 1 - m, 1 - j] \\ [c, m, j] &= [1 - r, 1 - v, 1 - b] \end{aligned} \quad (3.2)$$

Il est à noter que ces espaces ne permettent pas de caractériser une gamme de couleurs aussi large que celle que l'œil peut percevoir. Mais sachant que le médium entre la représentation interne et l'observateur (*i.e.* l'écran) utilise également cet espace, aucune autre couleur ne pourra être affichée, ce qui rend inutile toute tentative de représenter davantage de couleurs.

Espaces cylindriques. La famille des espaces cylindriques sépare la teinte H^3 et la saturation S d'une autre valeur, qui peut être la luminance, la clarté ou la luminosité. Elles s'appellent ainsi HSV , HSL , etc. Ces espaces sont populaires pour la définition de couleurs dans une interface graphique qui peut se faire de façon intuitive en définissant une teinte avant de régler la saturation et la luminance par exemple. Ces espaces sont dits cylindriques car la teinte décrit un cercle alors que la saturation l'étend à un disque et que la dernière composante varie selon l'axe de la longueur du cylindre (voir figure 3.6c). Ces coordonnées s'expriment en degrés pour la teinte et en pourcentage pour les autres composantes. En pratique, il prend une forme de cône car à luminance/clarté/luminosité nulle, seul le noir peut être représenté : plus cette composante est petite, moins le disque est grand.

La conversion entre RVB et un espace de cette famille est moins triviale. Voici, à titre d'exemple la conversion de (r, v, b) vers (h, s, L) :

$$\begin{aligned} Cmax &= \max(r, v, b) \\ Cmin &= \min(r, v, b) \\ \Delta &= Cmax - Cmin \\ h &= \begin{cases} 60^\circ \times (\frac{v-b}{\Delta} \bmod 6) & \text{si } Cmax = r \\ 60^\circ \times (\frac{b-r}{\Delta} + 2) & \text{si } Cmax = v \\ 60^\circ \times (\frac{r-v}{\Delta} + 4) & \text{si } Cmax = b \end{cases} \\ L &= \frac{\Delta}{2} \\ s &= \begin{cases} 0 & \text{si } \Delta = 0 \\ \frac{\Delta}{1-|2L-1|} & \text{sinon} \end{cases} \end{aligned} \quad (3.3)$$

3. Hue en anglais.

Espaces perceptuellement uniformes. Les espaces discutés ci-dessus ne sont pas *perceptuellement uniformes*. Le comité international d'illumination (CIE) a ainsi défini d'autres espaces ayant pour but de se rapprocher de l'uniformité perceptuelle. À ce titre ont été développés les espaces *CIE-XYZ*, *CIE-L*a*b**, *CIE-L*u*v**.

XYZ sépare la luminance Y de ses deux autres composantes X et Z décrivant la teinte. Ces derniers ont été définies de façon à ce que le calcul de distances en *XYZ* corresponde au mieux aux distances perçues par l'œil humain, sur base d'expérimentations menées autour de 1930 par *Wright* [1929] et *Guild* [1932]. L'espace *XYZ* a été créé pour se rapprocher de l'uniformité perceptuelle en se limitant à une transformation linéaire par rapport à *RVB* (voir figure 3.6d). Ainsi le résultat n'est pas parfait mais l'équation de passage sont simples :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M \begin{bmatrix} r \\ g \\ b \end{bmatrix} \text{ avec } M = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \quad (3.4)$$

M étant inversible, le passage inverse s'écrit :

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = M^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ avec } M^{-1} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \quad (3.5)$$

À noter que l'espace *xyY* en est une version normalisée, où les coordonnées *xy* représentent la teinte.

$L^*u^*v^*$ et $L^*a^*b^*$ sont le résultat de transformations non linéaires permettant de se rapprocher davantage de l'uniformité perceptuelle. Ces espaces ont été adoptés par le CIE en 1976. Elles séparent la luminosité L^* de la teinte dont elles ont chacune une représentation propre.

Les équations de passage entre $L^*u^*v^*$ et *XYZ* sont données par l'équation (3.6). Il suffira de les composer avec la transformation entre *XYZ* et *RVB* pour obtenir la transformation complète depuis l'espace *RVB*.

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_n}\right)^{1/3} - 16 & \text{si } \frac{Y}{Y_n} > 0.008856 \\ 903.3 \frac{Y}{Y_n} & \text{autour du noir.} \end{cases} \quad (3.6)$$

$$u^* = 13L^*(u' - u'_n)$$

$$v^* = 13L^*(v' - v'_n)$$

où

$$u' = \frac{4X}{X + 15Y + 3Z} = \frac{4x}{-2x + 12y + 3} \quad \text{et} \quad v' = \frac{9Y}{X + 15Y + 3Z} = \frac{9y}{-2x + 12y + 3} \quad (3.7)$$

et où (u'_n, v'_n) sont calculés pour un blanc de référence. Le blanc de référence D_{65} est souvent utilisé car il approche la lumière du jour. Sa valeur est $(X_n, Y_n, Z_n) = [0.950456, 1, 1.088754]$ ou $(u'_n, v'_n) = [0.197832, 0.468340]$.

L'espace $L^*a^*b^*$ est défini avec le même L^* que dans l'équation (3.6) et se définit par rapport à XYZ comme suit :

$$\begin{aligned} a^* &= 500 \left(f \left(\frac{X}{X_n} \right) - f \left(\frac{Y}{Y_n} \right) \right) \\ b^* &= 200 \left(f \left(\frac{Y}{Y_n} \right) - f \left(\frac{Z}{Z_n} \right) \right) \end{aligned} \quad (3.8)$$

où

$$f(t) = \begin{cases} t^{\frac{1}{3}} & \text{pour } t > 0.008856 \\ 7.787 t + \frac{16}{116} & \text{sinon} \end{cases}$$

La seule différence entre ces deux représentations réside en leur paramétrage de l'espace décrit par la teinte : les composantes a^* et b^* sont orthogonales. Alors que ces espaces sont actuellement utilisés dans le traitement d'image [Mittal *et al.*, 2012], des études poussées sont toujours en cours sur les différences intrinsèques entre ces espaces et leurs variantes [Thomas *et al.*, 2013]. La recommandation actuelle du CIE est d'utiliser l'espace $L^*u^*v^*$ pour la caractérisation de moniteurs de couleur et l'espace $L^*a^*b^*$ pour la caractérisation de surfaces colorées.

3.1.2 Comparatif

Le choix de l'espace de couleur va influencer qualitativement sur deux étapes de nos travaux :

- reconstruction : l'ajustement de fonctions aux données d'entrée issues des photographies se fera notamment avec une mesure d'écart entre couleurs reconstruites et mesurées ;
- simplification : celle-ci minimisera localement l'écart entre données denses et données simplifiées. Pour ce faire, un écart colorimétrique de radiance sera mesuré.

De plus, il y aura un coût de conversion en deux endroits si l'on choisit un espace autre que RVB :

- en amont de la reconstruction, une conversion transcrit les données dans l'espace désiré ;
- lors de la phase de visualisation où il faudra reconvertir les couleurs pour obtenir un affichage.

Le critère principal est ainsi l'*uniformité perceptuelle* car il s'agit bien de visualiser un objet tridimensionnel de façon à ce qu'il ressemble au mieux à l'original. Corréler la métrique d'écart à la perception humaine est donc primordiale.

La *complexité du calcul* importe peu lors de la phase de reconstruction car il s'agit de pré-calcul. Elle peut par contre être limitante dans la phase de visualisation car le calcul doit se faire à la volée.

Le *fondement physique* n'est pas un critère important. Un lien peut cependant exister

entre réalité physique et perception, mais ceci peut être considéré comme un effet de bord.

Le tableau 3.1 catégorise les familles d’espaces de couleur discutés ci-dessus selon les critères classés dans l’ordre d’importance.

Espace de couleur	Uniformité perceptuelle	Transformation
$L^*a^*b^*/L^*u^*v^*$	Très bonne	Non linéaire
XYZ	Bonne	Linéaire
RVB / CMJ	Faible	Linéaire
HSV	Très faible	Linéaire

Tableau 3.1 – Taxinomie et comparaison des espaces de couleur.

Nous pouvons dès lors immédiatement écarter

- l’espace CMJ , qui n’a aucun avantage calculatoire par rapport à RVB et qui n’est pas intéressant dès lors que nous n’utilisons pas d’imprimantes.
- les espaces cylindriques, qui n’apportent pas d’uniformité perceptuelle et ne représentent aucune réalité physique [Poynton, 1997].

Ainsi, les espaces $L^*u^*v^*$ et $L^*a^*b^*$ sont les mieux fondés de par leur uniformité perceptuelle, à condition que leur temps de calcul de conversion à la volée reste raisonnable. L’espace $L^*u^*v^*$ est plus approprié pour la synthèse additive de la couleur que nous exploitons lors de l’interpolation de couleurs sur le maillage [Wyszecki et Stiles, 2000]. Nous aurons donc tendance à préférer cet espace à $L^*a^*b^*$.

Nous avons établi un ordre de préférence mais ne faisons pas de choix *a priori*. L’utilisateur peut en effet avoir ses raisons propres pour choisir un espace de couleur : coût de conversion rédhibitoire sur son architecture, utilisation de bibliothèques supportant un espace en particulier, demandes de graphistes modélisant des objets, etc.

Ainsi, nous souhaitons fournir des solutions robustes à ce choix. Pour ce faire, nous avons implémenté l’espace tri-*stimuli* RVB et les espaces se rapprochant de l’uniformité perceptuelle : XYZ , $L^*a^*b^*$ et $L^*u^*v^*$. Nous fournirons des exemples de résultats représentatifs mettant en œuvre RVB et $L^*u^*v^*$ afin de démontrer l’indépendance de nos algorithmes à l’espace de couleur.

3.2 Modèles de fonction

Nous allons voir ici différents modèles connus permettant d’exprimer et de stocker les fonctions de radiance. Elles ont toutes un point commun : il s’agit de fonctions définies par une base et un ensemble de coefficients.

Vocabulaire

Modèle de fonction : le modèle de fonction comprend la base Φ , sa taille N et le paramétrage sur l'hémisphère \mathcal{H}_p (ou la sphère Ω).

Capacité de représentation : notion difficilement quantifiable définissant la complexité de radiance maximale représentable par le modèle de fonction choisi. Cette quantité est à considérer en proportion de la taille N .

Les critères suivants régissent souvent le choix d'une base :

- *Capacité de représentation* : la complexité de la radiance numérisée et le degré de réalisme souhaité.
- *Compacité* : les contraintes matérielles en termes de mémoire induisent une limite de capacité. La complexité mémoire (*i.e.* la quantité de coefficients à stocker par fonction) devra alors être limitée d'autant plus que le modèle 3D est grand.
- *Complexité d'évaluation* : les contraintes matérielles et logicielles (*i.e.* implémentation CPU ou GPU et puissance de calcul de ces derniers) et la taille des données influent sur la vitesse de rendu. La complexité d'évaluation d'une fonction dépend de la base et de sa taille.
- *Complexité de l'ajustement* : les coefficients seront estimés afin d'approcher les couleurs acquises. Selon la base choisie, on pourra avoir recours à un ajustement linéaire ou non.
- *Domaine de définition*. Certaines bases sont définies sur un domaine fermé (*e.g.* sphérique), d'autres sur un ouvert (*e.g.* hémisphérique, plan tangent). Selon ce choix, on sera contraint ou non d'exprimer la fonction dans un repère local plus compact mais nécessitant d'être orienté explicitement, contrairement à un repère global. Nous en verrons les enjeux au cours des chapitres à venir.

D'autres critères non scientifiques peuvent être importants pour un programmeur, à savoir la disponibilité d'outils dans des bibliothèques et la complexité d'implémentation.

Certains critères sont plus importants que d'autres selon les contextes. De plus, certains d'entre eux sont contradictoires. Il est donc impossible de toutes les satisfaire. Nous souhaitons laisser la liberté de choix à l'utilisateur et n'allons donc pas l'imposer *a priori*. Nous portons un soin accru à valider nos contributions sur tout type de modèle de radiance. Ainsi, nous rendons nos contributions robuste à ce choix.

3.2.1 Représentations analytiques

Dans la suite de cette section, nous présentons diverses bases issues de la littérature, dont toutes celles que nous mettrons en œuvre. Elles sont illustrées dans la figure 3.7.

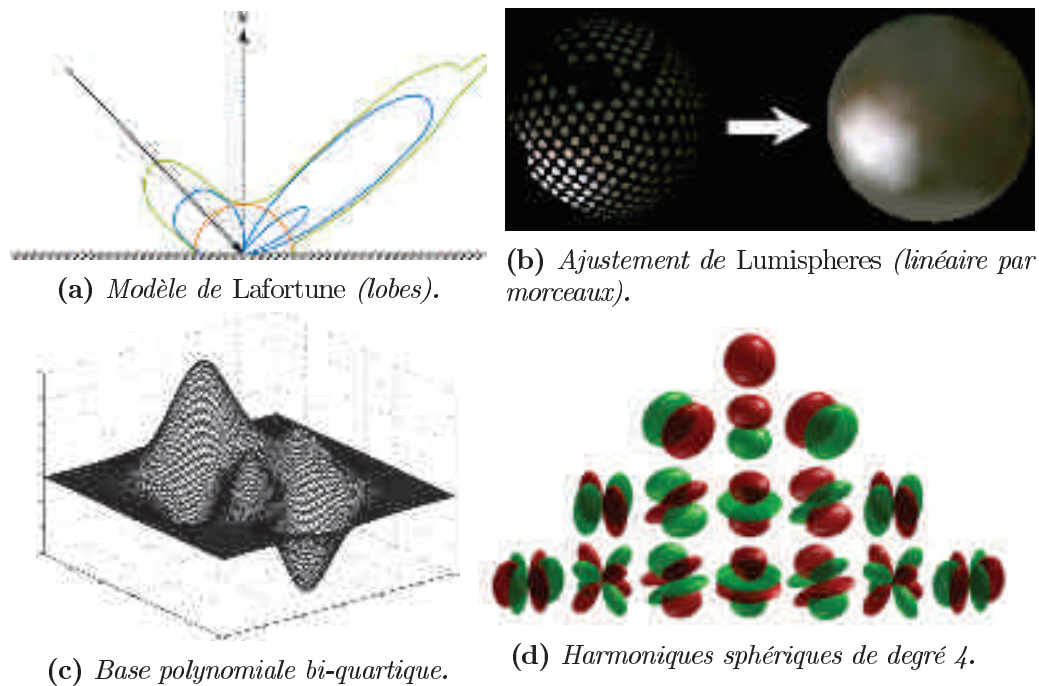


Figure 3.7 : Exemples de fonctions de radiance.

Modèle de Lafortune. Une première famille de fonctions découle de modèles permettant de représenter des BRDF. Il s'agit de variantes du modèle de Lafortune dont deux degrés de liberté (ceux de l'éclairage) sont fixés. Elles sont caractérisées par un cosinus élevé à une puissance pour représenter les lobes de réflexion spéculaire. Ils peuvent représenter des matériaux allant du type *Lambertien* à la réflexion spéculaire voire la rétro-réflexion. Cette famille englobe notamment les modèles de *Blinn-Phong*, *Cook-Torrance* et de *Lambert* vus précédemment [Phong, 1975; Cook et Torrance, 1982; Lafortune *et al.*, 1997].

Ces modèles ont l'avantage de pouvoir précisément représenter la réflexion de beaucoup de matériaux. Or, ces fonctions sont non-linéaires en leurs coefficients, ce qui complique leur ajustement et leur évaluation. Les représentations suivantes sont des bases de fonctions non basées sur des réalités physiques mais utilisées en pratique pour représenter des données issues d'une numérisation.

Lumispheres. Les *Lumispheres* [Wood *et al.*, 2000] sont la première représentation utilisée pour représenter des fonctions de radiance dans le cadre des champs de lumière surfaciques.

Il s'agit d'une base de la sphère – et non de l'hémisphère – linéaire par morceaux. La sphère a été régulièrement subdivisée en 258 afin de définir une fonction linéaire bi-variée par 3 coefficients en chaque région. $N = 774$ coefficients sont ainsi nécessaires par fonction, mais une compression efficace rend ce coût mémoire non prohibitif.

Il est à noter que la moitié de la sphère ne sera jamais visualisée. Or, il est important pour l'outil de compression que présente cet article d'avoir une définition omnidirectionnelle de la radiance. La quantification utilisée exploite en effet des similitudes entre *Lumispheres* correspondant à des orientations de surface différentes.

Bases polynomiales. Les bases polynomiales se trouvent dans les textures polynomiales (PTM⁴) [Malzbender *et al.*, 2001]. Il s'agit de textures dont chaque texel est décrit par une fonction hémisphérique polynomiale bi-quadratique dont le paramètre est la direction incidente ω_i d'une source d'éclairage ponctuelle. Elles permettent de ré-éclairer virtuellement des objets (acquis ou simulés) visualisés à partir d'un point de vue unique.

Ces fonctions sont mises en œuvre comme suit. Une paire de coordonnées (u, v) décrit l'hémisphère en parcourant le disque unitaire qui est le résultat de la projection orthogonale de l'hémisphère sur le plan tangent. Un ensemble de $N = 6$ coefficients suffit alors pour déterminer la fonction.

$$L(u, v) = [c_1 \dots c_6][1 \ u \ v \ uv \ u^2 \ v^2]^T \quad (3.9)$$

Nous utilisons ce modèle pour représenter de la radiance : u et v décrivent alors toujours un disque mais représentant l'hémisphère des angles sortants ω_r au lieu de ω_i . Nous étendons par ailleurs l'équation (3.9) à des degrés supérieurs, notamment bi-cubique (10 coefficients) et bi-quartique (15 coefficients). De façon générale, il y a $N = \frac{(d+1)(d+2)}{2}$ coefficients par polynôme (où d en est le degré).

Harmoniques sphériques. Les harmoniques sphériques [MacRobert, 1948] sont des polynômes paramétrés en coordonnées sphériques sur le domaine fermé qu'est la sphère. Il s'agit de la portion angulaire de la solution à l'équation de *Laplace* en coordonnées sphériques. La fonction s'exprime en nombres complexes mais admet une variante réelle qui s'écrit comme suit :

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l c_l^m K_l^m P_l^m(\cos\theta) \cos(m\phi) \quad (3.10)$$

où les c_l^m sont les coefficients de la fonction,
 où $K_l^m = \frac{2(l-m)!}{(l+m)!}$ est une constante,
 et où $P_l^m(x)$ est un polynôme associé de *Legendre*.

En pratique, le nombre de degrés l est fini et délimitera la capacité de la fonction à représenter des effets complexes. Il y a $N = (l+1)^2$ coefficients par fonction.

Une variante hémisphérique existe : elle ramène toutes les fonctions de base sur un hémisphère afin d'augmenter la capacité de représentation à nombre de coeffi-

cients constants [Gautron *et al.*, 2004]. Davantage de détails sur cette représentation peuvent se trouver dans la thèse d'Olivier GÉNEVAUX [2006].

Autres représentations. Dans la suite de ce manuscrit, nous évoquons les représentations que nous venons d'énoncer. Nous en implémentons certaines (les harmoniques sphériques et la base polynomiale) et discutons leurs différences dans la section 3.2.2.

Il en existe cependant d'autres.

Citons par exemple la base des ondelettes sphériques [Schröder et Sweldens, 1995] qui permet de représenter des fonctions sur la sphère. À l'instar de toute base d'ondelettes, elle permet d'exploiter la cohérence directionnelle pour diminuer le nombre de coefficients nécessaires à la représentation d'une fonction. Eu égard aux problématiques que nous discutons dans la section à venir, cette base exhibe des propriétés assez similaires aux harmoniques sphériques : forte capacité de représentation d'effets directionnels et domaine de définition fermé et sphérique notamment.

Dans [Jachnik *et al.*, 2012] finalement, l'hémisphère des directions est discrétisée régulièrement en un certain nombre de cercles concentriques du plan de projection orthographique (*i.e.* les parallèles sur l'hémisphère), typiquement 60. Sur chaque cercle, une résolution angulaire détermine un nombre de points pour lesquels une couleur est stockée. Il s'agit donc d'une représentation assez proche des *Lumispheres* (une représentation discrétisée) à cela près que le rendu se fait par interpolation de couleurs stockées, alors que pour les *Lumispheres* il s'agit d'évaluations de fonctions linéaires sans interpolation. La représentation de [Jachnik *et al.*, 2012] est donc assez proche des *Lumispheres* concernant les critères abordés dans la section à venir.

3.2.2 Comparatif

Toutes ces représentations de fonctions (à l'exception des dérivées du modèle de *Lafortune*) sont exprimables comme une combinaison linéaire des coefficients (voir équation (3.1)). Ceci délimite l'ensemble des fonctions sur lequel tous nos travaux réalisés s'appliquent et ont été testés.

Nous allons commenter la performance des différentes bases selon les critères susmentionnés. Nous n'avons pas fait d'étude comparative de cette complexité : les commentaires qui suivent découlent de la complexité théorique des équations des fonctions et/ou d'une expérience empirique de leur utilisation (nous évoquons une possible étude dans les perspectives). Le tableau 3.2 résume nos conclusions.

Capacité de représentation. La capacité de représentation est inhérente à la base utilisée et au domaine de définition des fonctions. Ainsi, des modèles à base de lobes (*i.e.* harmoniques sphériques et *Lafortune*) sont plus appropriés que

Base de fonctions	Ajustement aisé	Capacité de représentation	Compacité	Repère	Complexité d'évaluation
HS	Oui	Lobes	Correcte	Libre	Très forte
BP	Oui	Faible	Correcte	Local	Très faible
Lumisph.	Oui	Faible	Faible	Libre	Faible
<i>Lafortune</i>	Non	Spécularité	Bonne	Local	Forte

Tableau 3.2 – Comparaison des bases de fonctions.

les *Lumispheres* ou les bases polynomiales pour représenter des réflexions spéculaires. Les PTM se satisfont en effet d'une représentation incapable de représenter de fortes variations directionnelles : le ré-éclairage influe davantage sur le mouvement d'ombres que sur l'apparition de pics spéculaires. Les *Lumispheres* quant à eux ont été sur-échantillonnés dans ce domaine jusqu'à satisfaction du résultat au prix d'un coût mémoire plus élevé.

Nous ne savons *a priori* pas ce que contiennent les fonctions de radiance que l'on souhaite représenter : cela dépend grandement du matériau composant sa surface et de l'environnement lumineux. Nous verrons qu'en pratique on observe souvent un comportement spéculaire. Dans ce cas les bases permettant de représenter des lobes sont plus adéquates.

Compacité. La compacité découle de la quantité de coefficients nécessaires par fonction : la taille N de la base. Idéalement, il faudrait déterminer le coût mémoire d'une fonction à qualité constante mais ceci est complexe à réaliser (nous y reviendrons dans les perspectives). Nous nous basons donc sur notre expérience subjective.

Des résultats visuels satisfaisants sont obtenus pour nos numérisations au prix d'une quinzaine de coefficients pour la base des harmoniques sphériques et les bases polynomiales. Ceci varie selon les objets que l'on représente : les polynômes nécessitent davantage de coefficients lorsque l'on représente des objets hautement spéculaires.

Les *Lumispheres* ont un coût constant de 778 coefficients par fonction.

Les modèles de *Lafortune* et les ondelettes sphériques n'ont pas été implémentés.

Complexité d'évaluation. Une base polynomiale est aisée à évaluer, tout comme la base linéaire par morceaux des *Lumispheres*. Les fonctions de *Lafortune* sont un peu plus complexes à évaluer de par leur caractère non linéaire avec fonctions trigonométriques. Finalement, les *harmoniques sphériques* requièrent davantage de calcul par l'évaluation de fonctions trigonométriques et des *polynômes de Legendre* (voir équation (3.10), les constantes peuvent être pré-calculées si besoin).

Complexité de l'ajustement. L'ajustement des fonctions se fera par minimisation d'une erreur (voir chapitre 4). Nous verrons que ce problème est quadratique lorsque la fonction est exprimée dans une base (ses valeurs sont calculées par combinaison linéaire de ses coefficients) et il peut être écrit comme la solu-

tion d'une équation linéaire. Une décomposition en valeurs singulières ou une descente de gradient permet alors de calculer le résultat souhaité. L'ajustement est plus délicat si la base de fonctions n'est pas linéaire en ses coefficients (*e.g.* pour le modèle de *Lafortune*) : le problème est alors non quadratique.

Domaine de définition. Selon la base choisie, le programmeur est libre ou non de définir un repère global ou local. Nous verrons dans les chapitres 5 et 6 qu'un repère global facilite le calcul sur le maillage : comparaison de fonctions et interpolation notamment. La définition en repère local présente en effet de nombreux obstacles à cela : nécessité de définir de fastidieuses transformations de repères qui ne sont par ailleurs pas toujours possibles analytiquement.

La base polynomiale définit une projection de l'hémisphère sur un autre espace de coordonnées. Ceci peut induire des distorsions. Nous avons mené une étude qualitative comparant diverses projections : *orthogonale*, *authalique* et *conforme*. La projection *conforme* génère une projection uniforme, *i.e.* les aires sont préservées. Nous jugeons cependant peu utile de l'utiliser au vu des faibles différences que nous avons constatées : nous utilisons donc la projection orthogonale, nécessitant le calcul le moins coûteux.

4 Conclusion

Nous avons présenté la représentation du champ de lumière : un espace 4D définissant la couleur d'un objet ou d'une scène en fonction du point de vue. Au vu de notre cadre applicatif, nous souhaitons représenter un seul objet avec une grande finesse géométrique, c'est pourquoi nous avons choisi le champ de lumière surface. Dans ce modèle, les données acquises – un ensemble de pixels issus des photographies – sont regroupés par point d'émission de la surface. Dans l'approche locale que nous mettons en œuvre, une fonction de radiance est définie en chacun de ces points, dont la quantité est finie.

Nous avons déterminé 4 éléments qui peuvent varier lors du choix d'une représentation de fonctions de radiance : le point d'attache (sommet ou texel), le domaine (hémisphérique orienté localement ou sphérique orienté globalement ou localement), l'espace de couleur et la base de fonction.

Les enjeux des deux premiers seront détaillés en début de chapitre 5. Le point d'attache importe peu au chapitre 4 car les reconstructions de fonctions se font indépendamment d'un point à l'autre.

Nous verrons dans ce même chapitre que grâce à notre contribution de reconstruction robuste, ni le choix du domaine de définition, ni celui de l'espace de couleur, ni celui de la base ou encore de sa taille forme un obstacle pour la reconstruction. Ainsi, nous ne faisons aucun choix privilégié *a priori*. Au contraire, afin de démontrer la robustesse de nos contributions, nous nous donnons deux variantes permettant de faire varier le domaine parmi une représentation sphérique ou hémisphérique, paramétré en coordonnées sphériques ou cartésiennes, sur un domaine ouvert ou fermé :

- harmoniques sphériques paramétrés en coordonnées sphériques sur la sphère, donc sur un domaine fermé,
- base polynomiale paramétré en coordonnées cartésiennes sur le disque projeté orthogonalement à partir de l'hémisphère, donc sur un domaine avec bords.

En pratique, nous avons utilisé des harmoniques sphériques de degrés 0 à 19 et des bases polynomiales de degrés 2 à 4. La base polynomiale est obligatoirement locale.

Dans le chapitre 4, il s'agit de déterminer une fonction (*i.e.* ses coefficients) afin d'approcher les pixels directionnels acquis en entrée. Dans le chapitre 5, il s'agit de déterminer la couleur à rendre en tout point de la surface par interpolation des fonctions de radiance définies en une quantité finie de points. Finalement, au chapitre 6, il s'agit de simplifier le modèle de champ de lumière surfacique obtenu de façon à minimiser un écart à l'original.

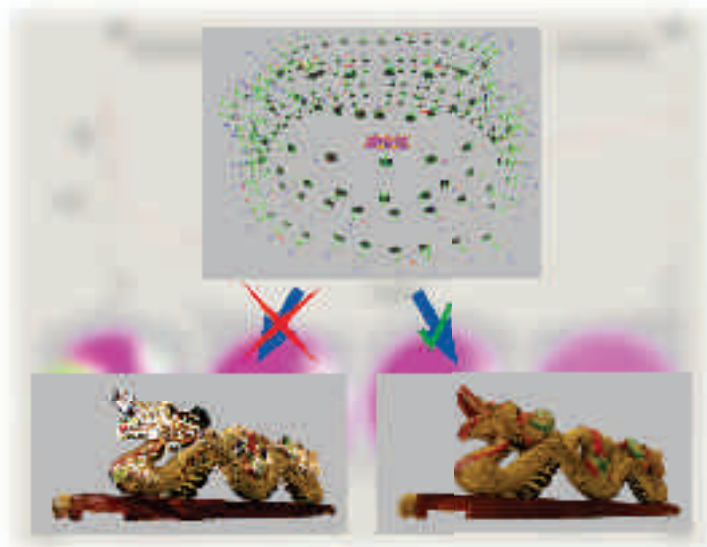
DEUXIÈME PARTIE

TRAVAUX RÉALISÉS

Chapitre 4

Reconstruction robuste de fonctions de radiance

Robuste (Larousse) : « Qui ne se laisse pas ébranler facilement »



RÉSUMÉ:

Ce chapitre aborde le procédé local qui consiste à estimer les coefficients d'une fonction de radiance de façon à approcher la radiance dont un échantillonnage a été acquis. En section 1, nous présentons un algorithme d'ajustement naïf qui permet d'ajuster une fonction sur les échantillons fournis. Or, les variables fournies en entrée (échantillonnage local, représentation de la fonction de radiance, bruit d'acquisition, etc.) sont *a priori* inconnus et peuvent pour certains varier d'un point à l'autre. Ainsi, l'algorithme naïf produit parfois des résultats non souhaités. En section 2, nous faisons le constat des causes des ajustements non souhaités et proposons de minimiser un écart à une énergie de stabilisation en complément de l'écart aux pixels directionnels. Ceci produit des résultats souhaités, quelque soient les conditions d'entrée. Dans la section 3, nous expliquons comment déterminer un compromis optimal entre proximité aux données fournies et robustesse de l'ajustement. Ceci se fait à l'aide d'une mesure quantitative basée sur un procédé statistique. En somme, on aboutit à un algorithme permettant de générer un résultat de qualité lorsque les conditions sont bonnes (bon échantillonnage, base de fonction adéquate, faible bruit) et un résultat « plausible » et sans artefacts sinon. Nous montrons également la ré-utilisabilité de l'algorithme dans le contexte de textures ré-éclairables.

Sommaire

1	Ajustement de fonction	75
1.1	Approcher les pixels directionnels	76
1.2	Choix de la métrique	77
2	Robustesse de la reconstruction	80
2.1	Défaut de robustesse	80
2.2	Stabilisation de l'ajustement	83
2.2.1	Choix de l'énergie de stabilisation	83
2.2.2	Comparaison théorique	84
2.3	Résultats et comparaison qualitative	86
2.3.1	Jeu de données	87
2.3.2	Propriétés fondamentales	90
2.3.3	Robustesse aux choix de l'utilisateur	92
2.4	Bilan de la stabilisation	94
3	Analyse statistique	96
3.1	Biais et variance	97
3.2	Compromis entre qualité et robustesse	98
3.3	Résultats quantitatifs	103
3.4	Solutions pratiques	104
4	Conclusion	106
4.1	Bilan	106
4.2	Perspectives	107
4.3	Ouverture aux textures polynomiales	108

Ce chapitre traite le passage de données issues des images aux fonctions de radiance. Après la rétro-projection, les données « image » sont rattachées à la surface en un ensemble fini de points que l'on traitera individuellement. En chaque p_i , un ensemble de pixels directionnels (*i.e.* couleurs associées à une direction ω_r définie sur l'hémisphère visible) forment un échantillonnage de la radiance. Nous ajustons une fonction de radiance continue sur chaque hémisphère. La fonction ajustée a vocation à approcher la radiance mesurée.

Vocabulaire

Échantillonnage de p_i : ensemble de K pixels directionnels $\{(\omega_r, v)_k = (\omega_r^k, v^k), k \in [1, K]\}$ rétro-projetés en p_i . Nous visualisons les échantillonnages sur un disque issu de la projection orthographique de l'hémisphère visible sur le plan tangent, comme par exemple dans les figures 4.1b, 4.2 ou 4.4a.

Ajustement : estimation des paramètres déterminant une fonction de radiance de façon à approcher l'échantillonnage fourni. Les fonctions ajustées sont également visualisées sur un disque, tel que dans les figures 4.2 ou 4.4b.

La difficulté que nous allons aborder est due à la grande variété d'échantillonnages. La figure 4.1 illustre l'objet « double dragon » et son acquisition photographique. La figure 4.1a montre l'emplacement des prises de photos par rapport à l'objet : ils sont nombreux et plutôt bien répartis dans un hémisphère au-dessus de l'objet (il est impossible de photographier l'objet d'en-dessous vu qu'il est posé sur un solide). La figure 4.1b montre que malgré cette bonne couverture photographique, les échantillonnages sont très variés : en fonction de l'orientation et des auto-occultations, le nombre K de pixels directionnels par hémisphère varie du simple au centuple.

Nous verrons dans la section 2 qu'ajuster des fonctions de radiance naïvement sur ces données (*i.e.* avec pour seule contrainte de minimiser l'écart aux pixels directionnels fournis) est problématique : le résultat est souvent inattendu voire imprévisible. Par exemple, la sous-constriction numérique de l'ajustement permet de générer plusieurs résultats interpolants différents. De même, une mauvaise répartition des échantillons dans le domaine hémisphérique peut générer un résultat non souhaité dans des parties non couvertes.

De nombreux paramètres connus influent sur le résultat :

- la taille de la base (*i.e.* le nombre N de degrés de liberté),
- le nombre d'échantillons (*i.e.* le nombre K de degrés de liberté contraints),
- la « nature » de la base (*i.e.* sa capacité de représentation),
- la répartition des échantillons,
- l'espace de couleur choisi, et
- le domaine de définition et son paramétrage.

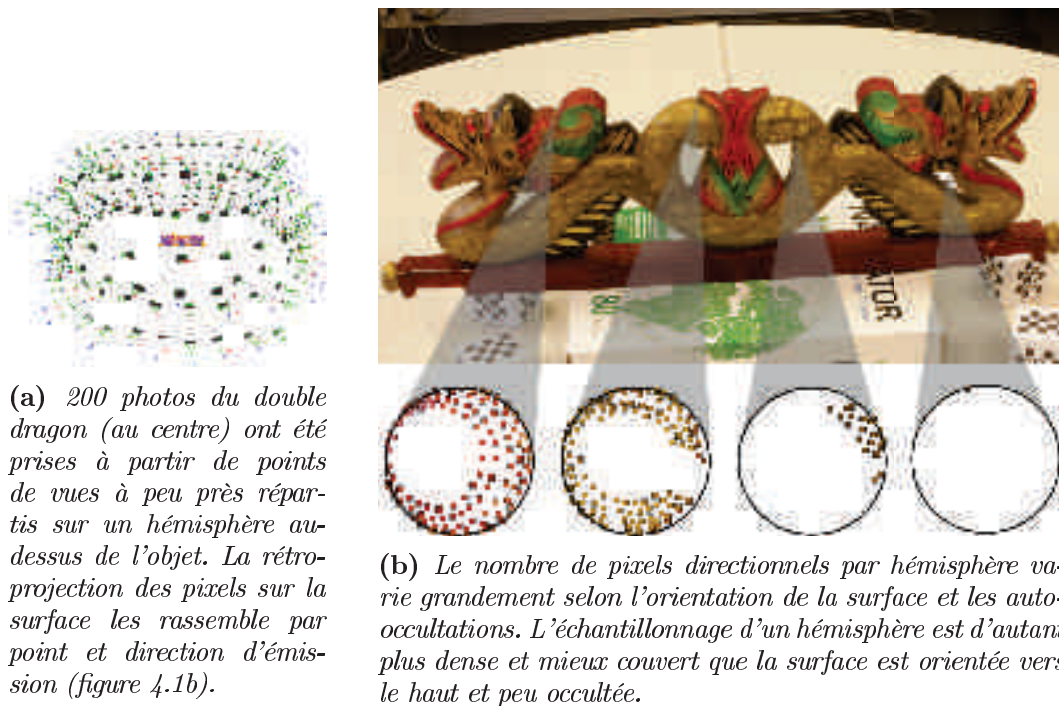


Figure 4.1 : Cas de figure souvent retrouvés dans une même campagne d'acquisition.

Il est cependant difficile d'en déduire une heuristique permettant de prédire le comportement d'un ajustement.

D'autres paramètres externes sont inconnus :

- Le bruit d'acquisition et de reconstruction. Il est le résultat d'une composition de nombreuses approximations dont le matériel, les algorithmes de reconstruction ou les limites de précision numériques sont responsables. En établir un modèle est ainsi exclu.
- Le modèle de radiance à représenter. Après acquisition, nous ne connaissons pas ce modèle mais uniquement un échantillonnage plus ou moins heureux de ce dernier. Nous ne disposons donc d'aucune « vérité terrain ».

Certains de ces paramètres sont quantifiables (*e.g.* taille de la base, nombre d'échantillons), mais d'autres sont difficiles à gérer : comment éliminer le bruit ou comment déterminer précisément la capacité de représentation de la base ou celle dont la radiance mesurée aurait besoin.

Dans ce chapitre, nous contraignons l'ajustement de fonction sur les pixels directionnels par la minimisation d'une énergie de régularisation. Ceci contraint numériquement l'ajustement dans un premier temps afin de générer un résultat prévisible et déterministe. Nous choisissons ensuite l'énergie la plus appropriée afin d'également générer un résultat souhaité quel que soient les conditions (base, taille, etc.).

Vocabulaire

Résultat imprévisible : résultat d'apparence aléatoire, fortement dépendant de variations minimales dans l'échantillonnage d'entrée.

Résultat prévisible ou déterministe : résultat que l'on peut prévoir, robuste au bruit et aux petites variations dans l'échantillonnage d'entrée.

Résultat inattendu : résultat surprenant, *e.g.* une fonction de radiance définissant des couleurs hasardeuses sensiblement différentes de celles de l'échantillonnage.

Résultat souhaité ou attendu : résultat subjectivement jugé comme adéquat, *i.e.* sans artefacts ni couleurs inattendues.

Précision : proximité d'un résultat par rapport aux couleurs données en entrée.

Stabilité : se dit de l'ajustement générant un résultat prévisible. Par abus de langage, on parlera de résultat stable lorsque l'ajustement qui y a mené est stable.

La section 1 présente l'algorithme naïf et évoque le problème de la métrique d'écart aux pixels directionnels qui guidera l'ajustement. Dans la section 2, nous listons plus formellement les causes des problèmes de robustesse et proposons une solution pour la génération d'un résultat prévisible et souhaité. Cet algorithme permet de stabiliser l'ajustement en réduisant de peu la précision. La section 3 explique l'analyse statistique que nous avons appliquée afin de déterminer un compromis optimal entre précision et stabilité.

1 Ajustement de fonction

L'ajustement de fonctions vise à déterminer les coefficients d'une fonction de radiance afin d'approcher l'échantillonnage de la radiance mesurée (*i.e.* les pixels directionnels). En chaque p_i , nous disposons d'une constellation de pixels directionnels. Nous utilisons trois fonctions, une par canal de l'espace de couleur choisi. La figure 4.2 illustre le procédé local à l'aide de cette représentation : séparation en un échantillonnage par canal, ajustement des fonctions et combinaison après évaluation des fonctions lors du rendu. Le passage des pixels à la fonction se fait donc une fois par point surfacique et par canal¹.

1. Dorénavant, nous ne distinguons plus « radiance d'un canal » et « radiance dans notre vocabulaire ».

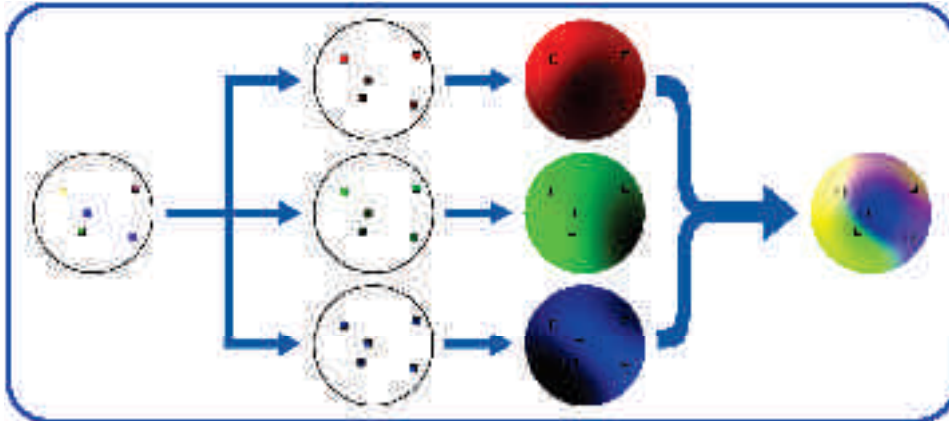


Figure 4.2 : Les pixels directionnels sont séparés en triplets (un par canal). Les fonctions hémisphériques à valeurs scalaires (une par canal) sont ajustées sur les pixels directionnels en entrée. Lors du rendu, les trois fonctions sont évaluées pour une direction donnée et combinées pour déterminer les trois composantes de couleur. Dans cette illustration, les fonctions sont évaluées en tout point de l'hémisphère.

1.1 Approcher les pixels directionnels

Pour une fonction donnée, il s'agit d'estimer les N coefficients $C^T = [c_1 \dots c_N]$ à partir de K pixels directionnels $\{\forall k \in [1, K], (\omega_r^k, v^k)\}$ fournis, *a priori* de façon à les approcher au mieux², de les interpoler et de les extrapoler :

$$R : \omega \mapsto C^T \Phi(\omega) \mid \forall k, R(\omega_r^k) = v^k \quad (4.1)$$

Ceci revient à résoudre le système suivant :

$$\begin{bmatrix} \Phi(\omega_r^1)^T \\ \vdots \\ \Phi(\omega_r^K)^T \end{bmatrix} C = \begin{bmatrix} v^1 \\ \vdots \\ v^K \end{bmatrix} \quad (4.2)$$

Une solution exacte de l'équation (4.2) n'existe pas en général (*e.g.* lorsque le problème est numériquement sur-contraint). Ainsi, une minimisation permet d'estimer les paramètres de la fonction telle qu'elle approche au mieux les pixels directionnels :

$$\arg \min_C E(C), \text{ avec } E(C) = \frac{1}{K} \left\| \begin{bmatrix} \Phi(\omega^1) \\ \vdots \\ \Phi(\omega^K) \end{bmatrix} C - \begin{bmatrix} v^1 \\ \vdots \\ v^K \end{bmatrix} \right\|_{L_p} \quad (4.3)$$

Inversement, si le problème est sous-contraint, plusieurs solutions sont exactes.

². Cette notion est volontairement floue à ce stade.

L'équation (4.3) est calculable avec une décomposition en valeurs singulières ou encore avec une des nombreuses variantes de descente de gradient (*e.g. Newton*, gradient conjugué), à condition que la norme p soit paire. Dans ce cas, l'équation (4.3) admet un minimum unique et global (ou plusieurs minima tous nuls) car le gradient de $E(C)$ étant

$$\nabla(E(C)) = \frac{p}{K} \left\| \begin{bmatrix} \Phi(\omega_r^1) \\ \vdots \\ \Phi(\omega_r^K) \end{bmatrix} C - \begin{bmatrix} v^1 \\ \vdots \\ v^K \end{bmatrix} \right\|^{p-1} \begin{bmatrix} \Phi(\omega_r^1) \\ \vdots \\ \Phi(\omega_r^K) \end{bmatrix} \quad (4.4)$$

la Hessienne de $E(C)$ est semi-définie positive :

$$H(E(C)) = \frac{p(p-1)}{K} \left\| \begin{bmatrix} \Phi(\omega_r^1) \\ \vdots \\ \Phi(\omega_r^K) \end{bmatrix} C - \begin{bmatrix} v^1 \\ \vdots \\ v^K \end{bmatrix} \right\|^{p-2} [\Phi(\omega_r^1) \cdots \Phi(\omega_r^K)] \begin{bmatrix} \Phi(\omega_r^1) \\ \vdots \\ \Phi(\omega_r^K) \end{bmatrix} \quad (4.5)$$

1.2 Choix de la métrique

Même lorsque l'échantillonnage est bon et que le problème d'ajustement est numériquement bien ou sur-contraint, il est utopique de penser qu'une reconstruction pourra en pratique être parfaite. Sa proximité aux échantillons dépend dans un premier temps du modèle de fonctions choisi par le concepteur ou programmeur, ce qui détermine la capacité de représentation des fonctions ajustées. Lorsque cette capacité est insuffisante pour représenter la radiance mesurée – rappelons qu'elle est induite par une réflectance du matériau et une illumination arbitraire – un biais est introduit lors de la reconstruction. Ce biais s'appelle le *biais modèle*.

Définition

Biais : écart, d'après une norme, entre échantillons et la fonction reconstruite.

Fonction interpolante : fonction ajustée avec un biais nul.

Biais modèle : tout ou partie du biais, à savoir la part due à un modèle à capacité trop réduite (*i.e.* une base de fonctions ne permettant pas d'interpoler les données).

En pratique, tous les exemples non synthétiques que nous montrons exhibent un biais modèle : plus d'une centaine de fonctions de base sont nécessaires ($N > 100$) pour représenter la radiance résultante d'une illumination générale (environnement extérieur par exemple), quelle que soit la base choisie (voir illustration en figure 4.13 à la page 95). Ceci n'est pas évaluable en temps-réel et/ou ne tient pas en mémoire GPU.

Exploiter le modèle choisi. Nous souhaitons exploiter au mieux la capacité de représentation finie qui aura été choisie par le programmeur pour représenter la radiance qui nous est inconnue. Minimiser l'écart aux mesures n'aboutit généralement pas au résultat souhaité : les reflets sont atténués ou alors source de fortes variations, générant respectivement un aspect diffus ou avec artefacts (nous en détaillons les causes en section 2).

Nous avons alors émis la double hypothèse que les radiances acquises sont le résultat d'une réflexion de type diffus + spéculaire et que les caractéristiques directionnelles (en particulier pics spéculaires) sont peu représentés au sein d'un échantillonnage car la probabilité de les acquérir est faible. Un ajustement avec la norme L_2 noierait alors les quelques pixels directionnels représentant les pics parmi les caractéristiques non directionnelles (la couleur diffuse) présents en nombre. C'est en ce sens que nous avons expérimenté avec un ajustement de la fonction selon l'équation (4.3) avec une norme L_p supérieure : $p > 2$. Nous avons choisi de comparer les normes L_2 , L_4 , L_6 et L_8 (la tendance se confirmera au-delà), qui renforce l'importance des valeurs extrêmes.

Expériences. Afin de vérifier l'impact de l'ajustement avec des normes supérieures, nous nous sommes placés dans le cas sur-contraint (qui générera un résultat déterministe). Nous avons simulé un éclairage et une réflectance sur des modèles géométriques connus (*i.e.* les modèles *vache*, *bimba* et *sphère* de la figure 4.3). Dans ce cas d'école, il s'agit d'un modèle de type diffus + spéculaire éclairé avec une source lumineuse blanche placée à grande distance de l'objet.

Sur la sphère, la réflectance ne s'applique que sur une moitié. L'autre moitié réfléchit selon un modèle *Lambertien*. Le modèle simulé constitue ici une vérité terrain dont nous avons acquis un échantillonnage directionnel régulier (sur la sphère englobante) et moyennement dense (une centaine d'acquisitions virtuelles). À partir de ces données, nous avons fait des ajustements de fonctions exploitant les différentes normes.

La figure 4.3 montre quelques résultats représentatifs des constatations empiriques que nous avons pu faire. Remarquons tout d'abord qu'aucune norme ne permet de représenter précisément la vérité terrain. Ceci est dû à l'ajustement sur un échantillonnage trop petit (*i.e.* la fréquence d'échantillonnage est insuffisante par rapport à celle du signal d'après le théorème de *Nyquist-Stokes*), mais aussi parce que le modèle polynomial utilisé (degré 4) ne permet pas de représenter ce modèle de radiance hautement spéculaire.

Sur le modèle de la vache, on peut voir que la norme L_4 a l'effet escompté : une importance accrue a été donnée au pic spéculaire dorsal. Cependant, ceci résulte en un éclaircissement dans toutes les directions d'observation. Ainsi, un éclaircissement est également observé sur des zones non spéculaires dans la direction d'observation courante (entre le cou et la tête par exemple).

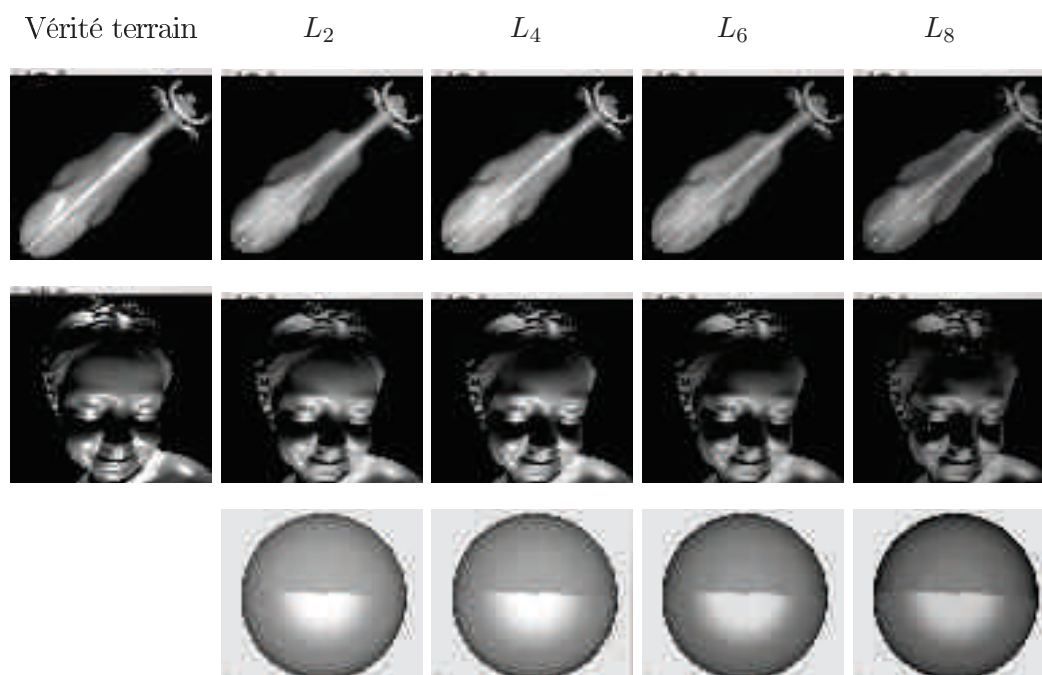


Figure 4.3 : Comparaison de l'ajustement de fonctions avec différentes normes sur trois objets. De gauche à droite : modèle simulé puis reconstructions respectives après ajustement utilisant les normes L_2 , L_4 , L_6 et L_8 . Les reconstructions exploitent une base polynomiale de degré 4 et l'espace de couleur RVB.

Augmenter la norme utilisée au-delà de 4 n'apporte rien voire dégrade le résultat :

- sur le modèle *bimba*, des artefacts importants apparaissent en des points qui sont sombres d'après le point de vue courant mais spéculaires d'après d'autres.
- sur le modèle sphérique, l'échantillonnage (plus particulièrement la visibilité d'un point à partir d'une caméra particulière) induit des effets en marche d'escalier amplifiés par la norme.

Limites. Les conclusions que nous venons de faire ne sont valables que dans nos cas particuliers : échantillonnage suffisamment dense pour éviter la sous-constriction. L'hypothèse sur la quantité de représentants de pics spéculaires parmi les pixels directionnels est en soi erronée : comme le montre la figure 4.1, il y a une telle variété d'échantillonnages que peu de certitudes existent à ce sujet. Nos analyses ne sont par ailleurs pas indépendantes de la base, ni du bruit. En particulier, l'imprévisibilité des résultats pour quelle norme que ce soit, nous a orienté vers la contribution traitant de la robustesse, présentée dans la section suivante.

Pour l'instant nous nous remettons donc à l'utilisation de la norme L_2 et évoquons la possible poursuite de cette étude dans la conclusion de ce chapitre.

Définition

E_{MSE} : l'énergie d'erreur quadratique moyenne est définie par l'équation (4.3) avec $L_p = L_2$.

Vocabulaire

Solution au sens des moindres carrés : résultat minimisant E_{MSE} .

Échelle macroscopique : l'échelle de l'objet entier (*e.g.* échantillonnage photographique global, visualisation d'un maillage).

Échelle mésoscopique : l'échelle d'une fonction de radiance (*e.g.* échantillonnage local, résultat de l'ajustement d'une fonction).

2 Robustesse de la reconstruction

Dans de bonnes conditions à l'échelle mésoscopique (notamment quantité et répartition des pixels directionnels), un ajustement au sens des moindres carrés donne des résultats « corrects ». Or, ces conditions découlent de phénomènes macroscopiques : couverture photographique et occultations notamment. Au vu du cadre qui est le notre (objets fragiles, de grande envergure ou faiblement accessibles), l'acquisition photographique se fait manuellement en plaçant « au mieux » l'appareil photo autour de l'objet. L'échantillonnage photographique est donc dans une certaine mesure *incontrôlée* et l'échantillonnage local d'un hémisphère peut être quelconque.

2.1 Défaut de robustesse

La méthode de minimisation de l'équation (4.3) est très sensible au conditionnement du problème d'ajustement. Il y a trois causes possibles à cela (pour rappel, un ajustement s'opère sur K échantillons et N est le nombre de degrés de liberté de la fonction à ajuster) :

1. La sous-constriction de l'équation d'ajustement (en général³ lorsque $K < N$) dû à une acquisition difficile ou aux auto-occultations générant une faible quantité de données.

Vocabulaire

Échantillonnage dense : $K \gg N$, l'ajustement est sur-contraint.

Échantillonnage creux : $K \approx N$, l'ajustement est souvent légèrement sur-contraint.

Échantillonnage très creux : $K \ll N$, l'ajustement est sous-contraint.

3. Ceci n'est exact que si la fonction est uni-variée. Dans notre cas bi-variée, cela dépend également du placement des pixels directionnels : si tous les pixels directionnels sont placés sur un axe du domaine de définition, l'autre ne sera jamais bien contraint.

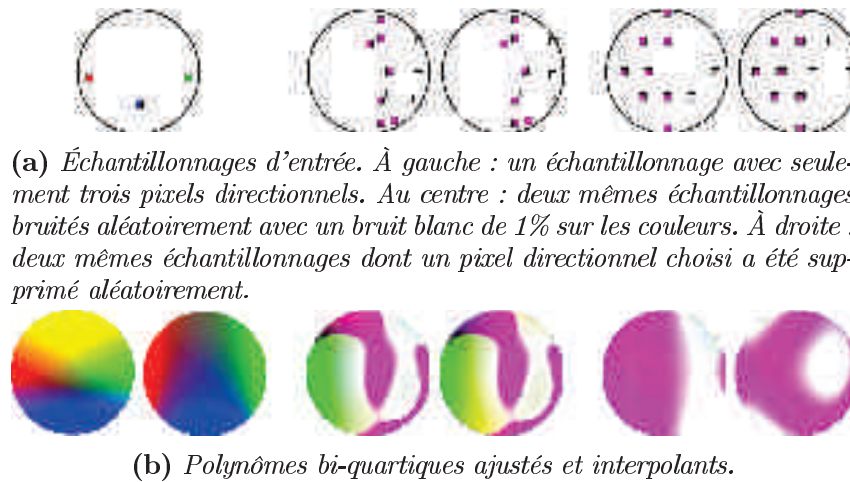


Figure 4.4 : De faibles variations en entrée causent des résultats très différents. À gauche : un échantillonnage sous-contraint ($N = 15$, $K = 3$) a plusieurs résultats interpolants possibles. Au centre et à droite : des échantillonnages faiblement différents génèrent des fonctions fortement différents.

2. La mauvaise répartition des directions d'observation acquises : une partie conséquente (potentiellement plus de la moitié) de l'hémisphère peut ne pas avoir été acquis, du fait des auto-occultations notamment.
3. Le bruit dans les pixels directionnels : tant la direction que la couleur est bruitée. Ce bruit inévitable peut être dû aux :
 - imprécisions dans les mesures géométriques : notamment dû au codage de la profondeur en nombres à virgule flottante ;
 - imprécisions dans les mesures photographiques : la quantification des couleurs (codage sur un octet par exemple) ainsi que le caractère discret du capteur ;
 - décalages entre le modèle géométrique reconstruit et le modèle réel, du fait des approximations algorithmiques et du compromis global assurant l'intégration de toutes les portions de surface issues de scans calibrés individuellement ;
 - erreurs de rétro-projection des pixels sur le maillage surfacique : un pixel couvre une portion de surface alors que sa couleur va être attachée à des éléments infinitésimaux ;
 - etc.

Souvent, le bruit est dû à tous ces éléments à la fois. La figure 4.1 donne un échantillon représentatif des cas survenant dans une même campagne. La figure 4.4 illustre quant à elle ces défauts de robustesse et leurs conséquences en termes de reconstruction de la fonction de radiance :

1. un même échantillonnage très creux admet plusieurs résultats interpolants différents,
2. deux échantillonnages faiblement bruités donnent des résultats sensiblement



Figure 4.5 : Ré-échantillonnage des directions d'observation (image de [Larue, 2008]). 1) les échantillons initiaux. 2) projection sur le plan tangent. 3) triangulation de Delaunay dans le disque projeté. 4) extrusion des bords de la triangulation pour couvrir tout l'hémisphère des directions d'observation. 5) génération de nouveaux échantillons régulièrement répartis dans le plan tangent, par interpolation bi-linéaire des valeurs aux sommets.

différents,

3. deux mêmes échantillonnages dont un pixel directionnel a été supprimé aléatoirement donnent des résultats sensiblement différents également.

En somme, les fonctions ajustées exhibent une *grande variance* par rapport au changement minime dans l'échantillonnage en entrée.

Définition

Algorithme d'ajustement robuste : ajustement dont la variance de la fonction ajustée est faible par rapport aux variations dans les échantillonnages.

Plus dérangent encore : la mauvaise répartition des pixels directionnels dans l'hémisphère laisse libre cours à la génération de *couleurs* inattendues. Ceci peut générer des artefacts des plus désagréables à visualiser. La figure 4.11 (à droite) sur la page 91 en est un exemple.

Nous nous trouvons dans la situation où l'apparition d'artefacts dépend de conditions d'échantillonnage non maîtrisées. Nous souhaitons au contraire que l'ajustement se comporte de façon robuste, résultant en une fonction de radiance

- ayant des couleurs *prévisibles* : aucune couleur non souhaitée doit apparaître ;
- peu dépendante de quel bruit que ce soit ;
- peu dépendant de l'ajout ou de la suppression d'un pixel directionnel (*i.e.* d'un point de vue occulté ou non).

La solution retenue pour régler ce problème dans la littérature est le ré-échantillonnage [Chen *et al.*, 2002; Larue, 2008] à partir des données disponibles (figure 4.5) : un ré-échantillonnage sur une grille régulière est déduit à partir d'interpolations et d'extrapolations bi-linéaires des données disponibles. Ainsi, le problème est bien contraint, ne générant qu'une seule solution minimisant l'équation (4.3). L'hémisphère est de plus bien couvert, inhibant l'apparition de couleurs hasardeuses.

Ceci est cependant au détriment de la qualité : une régularisation aura tendance à lisser les effets directionnels dans l'espace des directions d'observation. Instaurer

cette interpolation linéaire revient à ajuster une fonction linéaire par morceaux, puis à la ré-échantillonner avant d'ajuster une fonction de radiance. L'intérêt d'utiliser des fonctions plus complexes pour mieux représenter les données devient alors extrêmement réduit.

2.2 Stabilisation de l'ajustement

Nous proposons une autre approche pour résoudre les problèmes discutés ci-dessus : stabiliser le procédé en ajoutant une autre contrainte que l'attache aux données (équation (4.3)). Il s'agit d'une énergie dûment choisie et que nous appelons *énergie de stabilisation*.

$$\arg \min_C (1 - \lambda)E_{MSE}(C) + \lambda E_{stab}(C) \quad (4.6)$$

Ainsi, nous modifions le système de l'équation (4.3) en définissant une pondération entre attache aux données (minimisation de l'erreur quadratique moyenne E_{MSE}) et une autre énergie. Ceci permet dans un premier temps de résoudre le problème de sous-constriction numérique. Celle-ci ajoute une contrainte sur le « comportement » de la fonction. Alors que de nombreux choix peuvent résoudre le problème numérique, l'énergie devra également permettre d'éviter l'apparition de fortes variations dans les fonctions et de débordements de l'espace de couleur. L'objectif sera donc de choisir E_{stab} judicieusement, afin d'obtenir non seulement un résultat stable, mais également souhaitable. Dans cette section, nous comparons divers choix. Dans la section 3, nous voyons comment définir le compromis optimal entre attache aux données et stabilité : *i.e.* comment choisir λ dans l'équation (4.6).

2.2.1 Choix de l'énergie de stabilisation

Deux énergies ont été utilisées dans la configuration de l'équation (4.6) et pour l'ajustement de fonctions de radiance : l'énergie de plaques minces⁴, et l'énergie de la fonction. Il s'agit respectivement des dérivées d'ordre 2 et 0 de la fonction. Elles n'ont cependant pas été utilisées pour résoudre les problèmes que l'on évoque ci-dessus, mais pour des raisons qui leurs sont propres et que nous explicitons dans les paragraphes qui suivent. Ces énergies ne satisfont pas certains de nos objectifs qualitatifs, c'est pourquoi nous proposons d'utiliser l'énergie de *Dirichlet* : la dérivée d'ordre 1 de la fonction. Nous justifions ce choix dans cette section et illustrons les différences dans la section 2.3.

Nous présentons les énergies utilisées selon l'ordre de dérivation.

L'*énergie de la fonction* (dérivation d'ordre zéro) s'écrit comme suit :

$$E_{stab} = E_0 = \frac{1}{Aire(\mathcal{H})} \int_{\omega \in \mathcal{H}} \|R(\omega)\|^2 \quad (4.7)$$

4. *Thin-plate energy*, notamment utilisées dans les *thin-plate splines* [Wahba, 1990].

Dans [Lam *et al.*, 2006] elle a été utilisée dans le procédé d'ajustement d'*harmoniques sphériques* (définis page 64) représentant de la radiance. Le but est d'anticiper et de limiter le bruit introduit par la compression effectuée *a posteriori*.

L'*énergie de Dirichlet* (dérivation d'ordre un) se définit comme suit :

$$E_{stab} = E_1 = \frac{1}{\text{Aire}(\mathcal{H})} \int_{\omega \in \mathcal{H}} \|\nabla R(\omega)\|^2 \quad (4.8)$$

Nous suggérons son utilisation sur la base des objectifs suivants :

- pénaliser les fortes variations de couleur lorsque l'on s'éloigne (dans le domaine hémisphérique) des pixels directionnels acquis,
- pénaliser la sortie du domaine de l'espace de couleur,
- contrer le moins possible l'attache aux données.

Autrement dit, nous souhaitons que la fonction varie de façon à maximiser l'attache aux données, mais qu'elle soit plane (*i.e.* constante) lorsque l'on s'éloigne des pixels directionnels.

L'*énergie de plaques minces* (dérivation d'ordre deux) s'écrit dans un domaine paramétrique (x, y) comme suit :

$$E_{stab} = E_2 = \iint \frac{\partial^2 R(x, y)}{\partial^2 x} + 2 \frac{\partial^2 R(x, y)}{\partial x \partial y} + \frac{\partial^2 R(x, y)}{\partial^2 y} \quad (4.9)$$

Dans un domaine sans bord (*e.g.* une sphère), E_2 équivaut à $\frac{1}{\text{Aire}(\mathcal{H})} \int_{\omega \in \mathcal{H}} \|\Delta R(\omega)\|^2$, où Δ est l'opérateur de *Laplace-Beltrami* [Wahba, 1981]. Cette énergie fut utilisée dans [Wood *et al.*, 2000] pour ajuster des *Lumispheres* (définis page 63) sur des pixels directionnels. Ces fonctions paramétrées sur la sphère sont formées d'un ensemble de fonctions linéaires par morceaux à support compact (un triangle de la subdivision régulière de la sphère). Malgré le système d'acquisition rigide utilisé, certains triangles ne contiennent pas de pixels directionnels du fait des occultations. L'application de la contrainte de minimisation de E_2 sur la sphère permet de régler le problème de la sous-constriction locale. Il s'agit d'une régularisation dans le domaine sphérique. La figure 3.7b illustre qu'une fonction est définie même dans des zones non pourvues de pixels directionnels. Notons que ce problème n'existe pas avec une représentation à support non compact (HS et BP notamment).

Implémentation. Les intégrales des formules (4.7), (4.8) et (4.9) peuvent être calculées numériquement ou analytiquement. Nous les calculons analytiquement en tant que forme quadratique convexe par rapport aux coefficients par une décomposition en valeurs singulières.

2.2.2 Comparaison théorique

Ces trois énergies contraignent l'ajustement de fonction de façon à obtenir un résultat prévisible et donc robuste aux échantillonnages variants. La sous-constriction

numérique est évitée :

- dans tous les cas si l'énergie E_0 est utilisée (une fonction constante et nulle est alors générée),
- si au moins un pixel directionnel est fourni dans le cas de l'utilisation de l'énergie E_1 (une fonction constante de la couleur de ce pixel est alors générée),
- si au moins trois pixels directionnels non colinéaires sont fournis dans le cas de l'utilisation de l'énergie E_2 (une fonction linéaire interpolera alors ces trois valeurs).

Ne pas gérer le cas où aucune donnée n'est fournie n'est pas pénalisant : on voudra alors avoir recours à une fonction constante établie « par défaut ». Nécessiter trois pixels directionnels pour générer un résultat prévisible peut cependant être un handicap : il arrive souvent que certains échantillonnages ne contiennent qu'un ou deux pixels directionnels (voir figure 4.1b).

Dans un second temps, il s'agit d'obtenir un comportement souhaité. Un éclairage davantage intuitif du résultat généré selon l'énergie utilisée est de penser aux énergies par leurs propriétés physiques, illustrées dans la figure 4.6.

- L'énergie E_0 cherche à tirer la fonction vers 0.
- L'énergie E_1 cherche à ajuster une fonction constante sur les données.
- L'énergie E_2 cherche à ajuster une fonction linéaire (*i.e.* un plan en 3D).

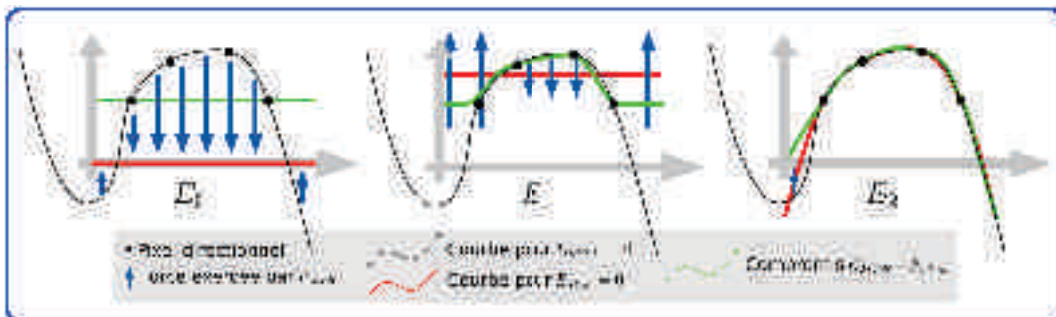


Figure 4.6 : Propriétés et tendances de la régularisation avec les énergies E_0 , E_1 et E_2 illustrées sur un cas 2D. Une fonction polynomiale de degré 3 est ajustée aux moindres carrés (courbe noire). Les flèches montrent la « force » exercée par la minimisation de l'énergie de stabilisation sur cette courbe. La courbe rouge est d'énergie de stabilisation nulle et constitue le but recherché par ces forces. La courbe verte montre un exemple de compromis.

Le désavantage de l'énergie E_0 dans notre contexte est précisément qu'elle tire les valeurs vers 0. Ceci représente en général du noir. Dans les parties d'hémisphère peu pourvues de pixels directionnels, et donc où l'attache aux données est faible, ceci fait rapidement (*i.e.* même pour λ petit) des dégâts. Les figures 4.9a, 4.10b, 4.11c et 4.12 illustrent cette tendance à générer du noir.

Ce problème est contré dans l'article de Lam *et al.* en se fixant une énergie cible basée sur un modèle de bruit. Un procédé de raffinement itératif alterne à chaque étape entre attache aux données et énergie cible. Le but de E_0 est ici non pas de tirer la fonction vers zéro, mais de la tirer vers une constante prédéfinie, ce qui pourrait

représenter une couleur diffuse. Dans notre cas, il est difficile de fixer une telle cible car les échantillonnages peuvent être creux.

L'énergie de *plaques minces* pénalise l'apparition de fortes variations. Cependant, lorsque peu de pixels directionnels sont présents au bord du domaine de paramétrage⁵, des couleurs extrapolées surprenantes peuvent apparaître car $E_2 = 0$ pour toute fonction linéaire. La figure 4.9c illustre ce problème : alors que tous les pixels directionnels en entrée sont gris et verts, l'apparition de magenta n'est pas pénalisée. Renforcer la stabilisation (*i.e.* augmenter λ) n'y fait rien car les couleurs sont représentées par une fonction linéaire ayant une énergie E_2 nulle.

E_1 ne souffre pas des problèmes de ces deux énergies. Elle cherche à déterminer une fonction dont les valeurs restent proches des couleurs en entrée. Elle évite ainsi de générer du noir (ou toute autre couleur inappropriée) et n'est pas sujette aux erreurs d'extrapolations.

Le seul bémol par rapport à l'énergie E_2 est qu'elle pénalise la transition qui vise à interpoler deux pixels directionnels voisins, alors que E_2 favorise les transitions linéaires.

Dans la section suivante, nous comparons les résultats sur des cas de test et montrons que l'énergie E_1 est la seule à fournir un résultat attendu et souhaité dans tous les cas.

2.3 Résultats et comparaison qualitative

Nous allons illustrer les effets produits par l'introduction de ces énergies sur un jeu de données varié. Toutes les énergies rendent le résultat robuste aux faibles variations dans l'échantillonnage d'entrée. Cependant, nous allons montrer que l'énergie E_1 est la seule produisant un résultat prévisible et cohérent lorsque l'on modifie

- l'espace de couleur,
- la base de fonctions,
- la taille de la base de fonctions,
- le domaine de définition et son paramétrage.

Nous verrons dans la section 3 à venir comment régler le paramètre λ de façon optimale. Si nous ne précisons pas la valeur de λ dans les comparaisons qualitatives de cette section, alors il s'agit du résultat optimal au sens de la définition que nous en donnerons. Celle-ci est différente selon l'énergie utilisée.

5. Ceci suppose que le domaine a un bord. Nous y reviendrons en section 4.

2.3.1 Jeu de données

Nos expérimentations portent sur trois types de données différents (figure 4.7). Elles montrent chacune différentes propriétés du processus d'ajustement.



Figure 4.7 : *Jeu de test.*

Échantillonnages de radiance synthétiques. Nous avons créé des échantillonnages en plaçant des pixels directionnels « à la main » sur l'hémisphère (figure 4.7a). Nous pouvons étudier le comportement de l'ajustement de fonctions à l'échelle mésoscopique en visualisant les fonctions générées. Les différents échantillonnages ont été créés afin de correspondre aux propriétés du début de la section 2.1 (variation des répartitions, quantité et bruit dans les échantillons). Il s'agit d'étudier l'apport des énergies de stabilisation.

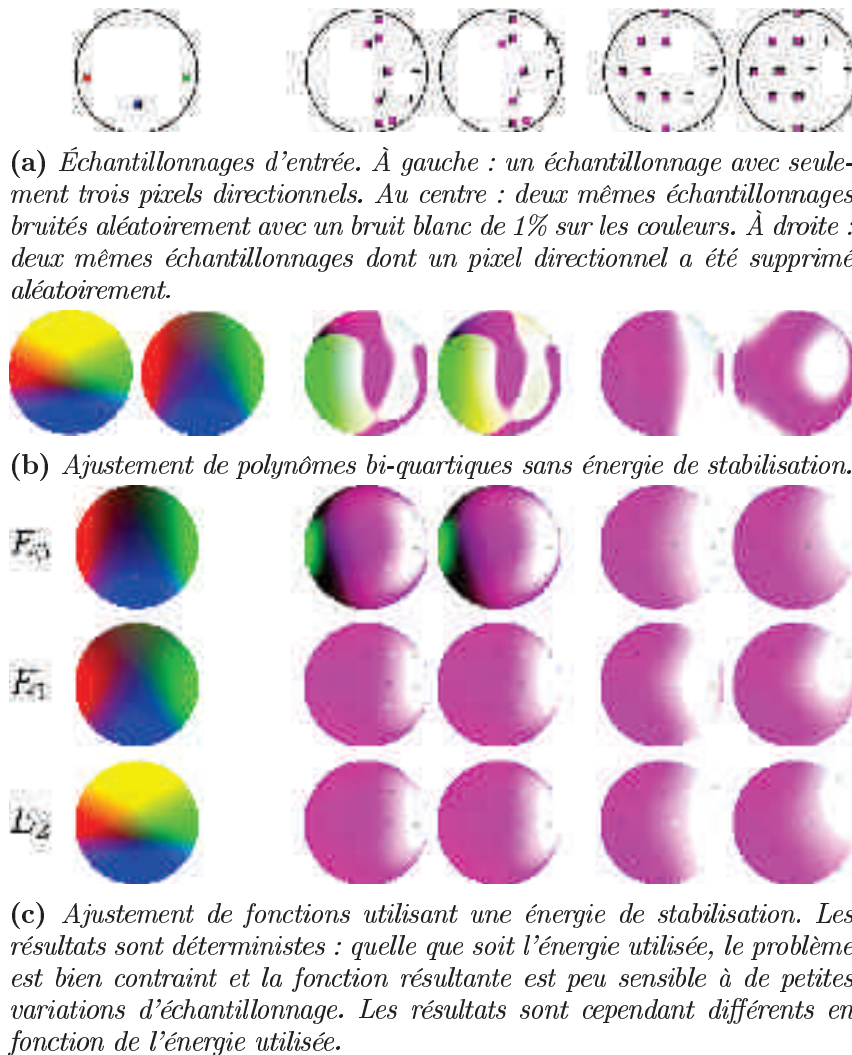


Figure 4.8 : Exemples d'ajustements de polynômes bi-quartiques déterministes.

Ces échantillonnages seront par ailleurs utilisés pour l'analyse statistique (section 3) afin d'aboutir à des données quantitatives dans un environnement connu.

Champs de lumière surfaciques synthétiques. Nous nous sommes donnés une géométrie connue et avons simulé un éclairage et une réflectance de la surface. Grâce à cela, on connaît la radiance de l'objet et on peut simuler une acquisition photographique afin de générer des données de champ de lumière synthétiques à l'échelle macroscopique. Ceci permet de contrôler l'environnement lumineux et d'éliminer le bruit d'acquisition, tant géométrique que colorimétrique (figure 4.7b).

Dans ce manuscrit, nous montrons le modèle *Stanford bunny* sur lequel on a simulé un éclairage par trois spots lumineux de couleur respective rouge, verte et bleue. La

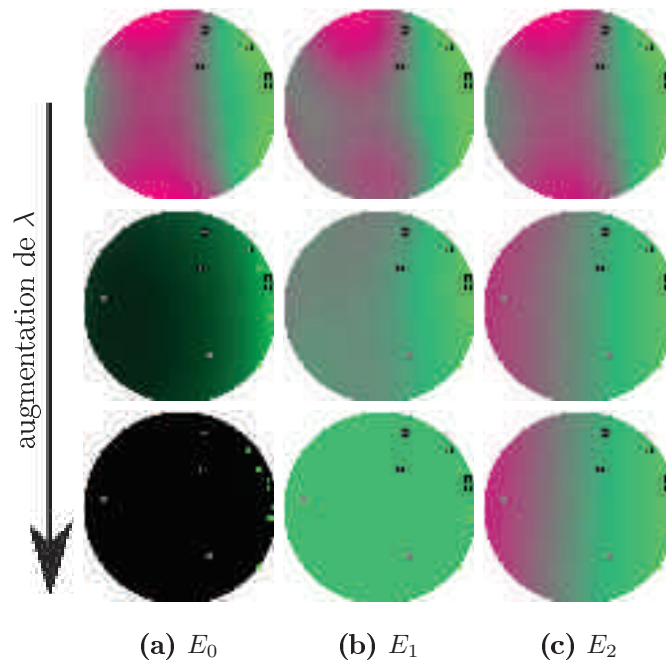


Figure 4.9 : Comparaison des énergies E_{stab} à l'échelle mésoscopique. Les mêmes pixels directionnels gris et verts sont fournis en entrée dans les trois cas. E_0 , E_1 et E_2 se comportent différemment lorsque λ grandit.

réflectance simulée est un modèle de *Blinn-Phong* et l'acquisition virtuelle de photos a été faite par un procédé de *Monte Carlo*. Ces spots et la forte réflectance simulée du matériau ont été choisis et placés afin de pouvoir montrer un comportement extrême où trois couleurs totalement différentes peuvent figurer dans une même fonction de radiance⁶.

51 photos virtuelles ont été acquises à partir de positions non uniformément réparties autour de l'objet. En moyenne, on a 11 pixels directionnels par hémisphère. 6,2% des hémisphères en ont moins de 5.

Champs de lumière surfaciques numérisés. Deux modèles ont été acquis : le *double dragon* et le *masque* (figure 4.7c). L'acquisition photographique fût réalisée manuellement, en parcourant approximativement un hémisphère au-dessus de l'objet. Ces données soulignent le besoin de stabilisation : de nombreux échantillonnages hémisphériques sont pauvres en pixels directionnels à cause des occultations de l'objet ou par le sol rendant une bonne partie de l'espace inaccessible. Prendre davantage de photos ne résout pas ce problème.

Le modèle du *double dragon* est intéressant pour ses caractéristiques de finesse va-

6. Ce cas pourrait être rencontré dans des numérisation réelles : un environnement lumineux complexe peut être assimilable à des spots de différentes couleurs.

riées, tant géométriquement (courbes grossières et détails fins) qu'en termes de radiance (diverses couleurs et propriétés de réflectance). La photométrie a été acquise par 200 photos. Le maillage surfacique fait 1,8 millions de sommets auxquels sont attachés les fonctions de radiance. En moyenne, 81 pixels directionnels sont rétro-projetés en un hémisphère, mais 5,5% des hémisphères en obtiennent moins de 20.

Le modèle du *masque* illustre un cas d'acquisition photographique massive : 672 photos ont été prises. Le maillage a été paramétré afin d'y définir une texture et de stocker les fonctions de radiance dans les texels : 2,8 millions de texels ont servi à représenter la face avant du masque.

2.3.2 Propriétés fondamentales

Nous analysons ici de façon générale les effets que produit la stabilisation sur le résultat d'un ajustement. Ensuite, nous comparons l'effet des trois énergies respectives.

Quelle que soit l'énergie utilisée, une stabilisation est effective : les résultats sont peu sensibles aux variations minimales dans les données d'entrée. Ceci est illustré par la figure 4.8, ainsi que par les figures 4.10 et 4.11. Les résultats sont fortement aléatoires d'un échantillonnage à l'autre lorsqu'aucune stabilisation n'est utilisée (voir 4.10a et 4.11b) et déterministe dans le cas contraire. Le résultat est néanmoins différent selon l'énergie utilisée.

L'énergie E_0 génère comme prévu un assombrissement de la radiance reconstruite dans les parties hémisphériques dépourvues de pixels directionnels. Ceci peut être observé sur les données synthétiques : les fonctions de radiance (figure 4.9a) indiquent la tendance et de nombreux artefacts sont produits sur le *bunny* (figure 4.10b) où des tâches noires sont rendues en des points peu photographiés. Le constat est moins flagrant sur les données réelles, mais l'assombrissement est également visible sur la figure 4.11c visualisé à partir de directions d'observation non photographiés (les vues « d'en dessous »).

L'énergie E_1 permet la reconstruction la plus convaincante. Les couleurs acquises (*i.e.* les pixels directionnels) sont correctement reconstruites et bien interpolées. Les différences avec l'énergie E_2 sont minimales sur les données réelles (par exemple en comparant les figures 4.11d et 4.11e). Les données synthétiques – construites dans le but de montrer ces différences – montrent cependant que l'énergie E_1 ne souffre pas du défaut d'extrapolation dans les parties non couvertes de l'hémisphère (figure 4.8c première colonne, figure 4.10c et 4.12 deuxième ligne). Les couleurs générées en toute direction (*i.e.* en tout point du disque) font partie des couleurs *souhaitées*.

L'énergie E_2 permet aussi une bonne interpolation entre les pixels directionnels (figures 4.8c à droite et 4.11e). Elle est cependant propice aux problèmes d'apparitions de nouvelles couleurs : une fonction linéaire n'est jamais pénalisée,

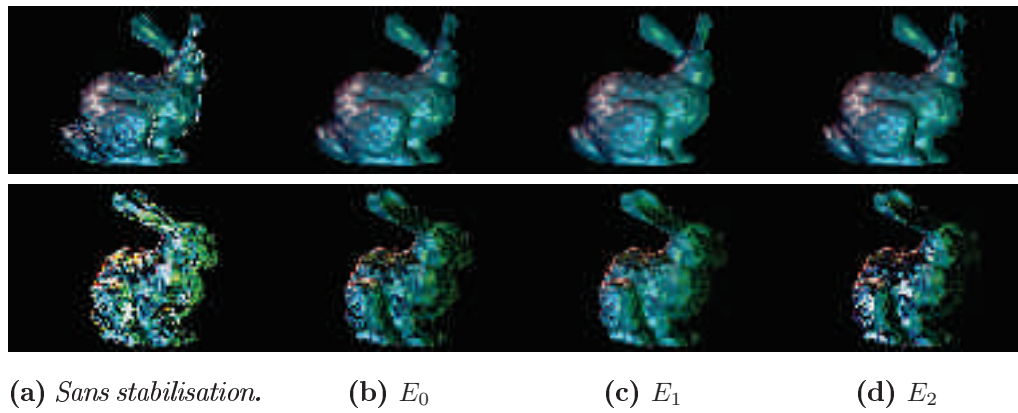


Figure 4.10 : Modèle du Stanford bunny : comparaison des énergies minimisées.



Figure 4.11 : Modèle du double dragon : comparaison des énergies minimisées.

alors qu'elle peut créer de nouvelles couleurs inattendues et non souhaitées. La figure 4.8c à gauche montre l'apparition de jaune. La figure 4.10d montre l'apparition de couleurs extrêmes : il s'agit souvent de valeurs numériques générées par la fonction de radiance qui sortent des bornes de l'espace de couleur (souvent du noir ou du blanc ou des couleurs saturées).

Pour résumer, les énergies E_1 et E_2 stabilisent le procédé et génèrent en règle générale de bons résultats. Nous avons cependant pu mettre en exergue un comportement déroutant de E_2 qui nous fera préférer l'utilisation de E_1 . Ci-dessous, nous proposons une analyse de la robustesse par rapport aux différents choix du modèle (espace de couleur, base de fonctions et taille de celle-ci) ainsi que du nombre de photos acquises.

2.3.3 Robustesse aux choix de l'utilisateur

Nous avons comparé les comportements des variantes de stabilisation par rapport aux choix de représentation et de prise de photos, à savoir :

1. l'espace de couleur,
2. la base de fonctions,
3. la taille de la base de fonctions,
4. le nombre de photos acquises.

Force est de constater que l'utilisation de l'énergie E_1 se démarque par son uniformité.

1—Espace de couleur. La figure 4.12 montre les variantes comportementales selon l'espace de couleur choisi. Les colonnes 4.12a et 4.12c illustrent le résultat pour une même base (HS) avec deux espaces de couleur différents. Les colonnes 4.12b et 4.12d font de même pour une autre base (BP). On constate que toutes les énergies se comportent de façon assez homogènes.

Cependant, l'énergie E_1 est la seule à créer des fonctions qui maintiennent les couleurs proches de celles acquises (en opposition à tendre vers le noir ou vers une extrapolation linéaire).

L'espace $CIE-L^*u^*v^*$ est propice aux problèmes de déphasage des canaux. Il vit en effet dans un sous-espace plus vaste que $[0, 1]^3$ et non homogène d'un canal à l'autre. Les fonctions varient donc plus fortement dans cet espace et il est ainsi plus difficile de faire passer le seuil de la couleur noire au même endroit sur l'hémisphère pour les trois fonctions représentant les différents canaux. Un tout petit écart induit l'apparition de couleurs lors du passage de ce seuil, visible dans la figure 4.12c pour E_0 et la figure 4.12d pour E_0 et E_2 . Pour corriger cela, la pondération entre attache aux données et stabilisation devra être adaptée canal par canal. L'énergie E_1 maintient

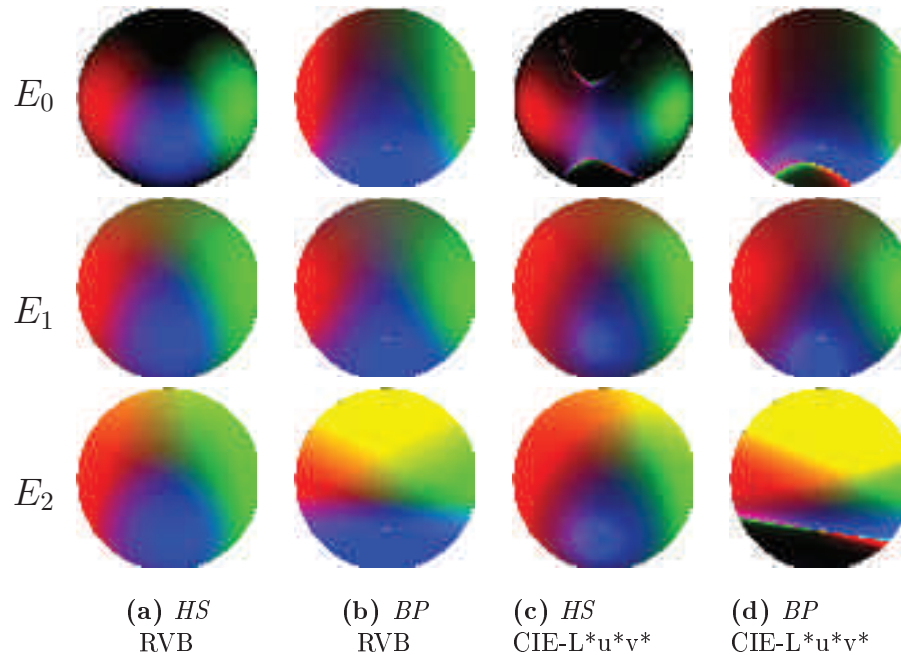


Figure 4.12 : Évaluation de la robustesse au choix de l'espace de couleur et à la base de fonctions. Seul trois pixels directionnels (respectivement rouge, vert et bleu) sont fournis. Les fonctions sont définies dans deux bases (HS de degré 3 et BP de degré 4) et deux espaces de couleur (RVB et CIE-L*u*v*).

mieux les fonctions dans le domaine de définition des couleurs, quelque soit l'espace utilisé. Ceci évite l'apparition de ces artefacts.

2–Base de fonctions. La même figure 4.12 montre également l'homogénéité de E_1 lorsque l'on modifie la base de fonctions utilisée. Les colonnes 4.12a et 4.12b montrent des résultats d'ajustements de fonctions avec des bases différentes et à espace de couleur constant (ici *RVB*). Il en va de même pour les colonnes 4.12c et 4.12d (avec l'espace de couleur $L^*u^*v^*$).

Les fonctions résultantes divergent grandement lorsque les énergies E_0 ou E_2 sont utilisées.

L'énergie E_0 souffre de l'apparition de couleurs noires de façon plus ou moins prononcée selon la base utilisée. La forme en « lobes » des harmoniques sphériques fait décroître la valeur de la fonction dès que l'on s'éloigne d'un échantillon fourni. Cette décroissance est plus limitée pour la base polynomiale.

L'énergie E_2 souffre plus ou moins du problème d'extrapolation selon la base choisie : les harmoniques sphériques s'exprimant sur un domaine sans bord, l'effet d'extrapolation est plus limité que pour les polynômes exprimés dans le plan tangent.

Finalement, même si de faibles divergences existent avec l'utilisation de E_1 selon la base utilisée, il est difficile d'affirmer que la moindre des solutions proposées soit

inattendue ou mauvaise : toutes les solutions sont « acceptables ».

3–Taille de la base de fonctions. La figure 4.13 illustre l'utilisation de deux tailles de base de fonctions pour la reconstruction d'une même acquisition : harmoniques sphériques de degré 9 ou 19 ayant respectivement 100 ou 400 coefficients par fonction. Le modèle 3D avec cette dernière taille de base ne peut pas être rendu en temps réel car il ne tient pas en mémoire GPU et nécessite ainsi du calcul hors ligne.

L'énergie E_1 se comporte de façon similaire lorsque la taille de la base (et donc le nombre de degrés de liberté) sont modifiés. On peut observer que le biais est quasiment nul dans la figure 4.13 à gauche (le biais non nul en quelques points suivant les motifs est causé par des erreurs de recalage) et assez faible à droite. Dans les deux cas, un résultat souhaité est obtenu. La même valeur de pondération (*i.e.* le paramètre λ) a été utilisé pour ces deux modèles. Ce paramètre peut être affiné selon la taille de la base de fonction choisie afin d'obtenir de meilleurs résultats. L'analyse que nous présentons dans la section 3 permet de répondre à ce besoin.

4–Nombre de photos acquises. La figure 4.14 montre une reconstruction (exploitant les harmoniques sphériques de degré 19) du masque à partir de sous-ensembles de photos acquises de plus en plus petit (variant de 672 à 11 images). Les deux photos à gauche correspondent aux points de vue reproduits à droite. Celle du haut a été utilisée dans toutes les reconstructions. Celle du bas n'est utilisée que dans les reconstructions basées sur 672 et 336 images.

Cette figure illustre ce que l'on peut attendre de la part de l'algorithme de reconstruction stabilisé lorsque le nombre de photos acquis varie. La ligne du bas montre qu'un résultat déterministe et souhaité est produit en tout état de cause, c'est-à-dire même si peu de photos sont utilisées et/ou si le point de vue visualisé n'est pas inclus dans l'échantillonnage. Le haut de la figure montre inversement qu'un résultat très précis peut être obtenu à condition de s'en donner les moyens (base large et échantillonnage photographique dense) mais avec le même algorithme d'ajustement.

2.4 Bilan de la stabilisation

L'ajout d'énergies de stabilisation dans le processus d'ajustement forme une solution aisée au problème numérique lié à l'ajustement de fonctions. Le problème est en effet bien contraint dès qu'un (respectivement trois) pixels directionnels sont fournis dans le cas de l'énergie E_1 (respectivement E_2). De plus, une solution stable, à savoir faiblement dépendante de variations mineures dans l'échantillonnage d'entrée, est fournie par toutes les énergies. Elles divergent cependant chacune par leur coût en termes d'attache aux données et de réduction d'artefacts.



Figure 4.13 : Reconstructions du masque avec différentes tailles de la base d'harmoniques sphériques : $N = 400$ à gauche et $N = 100$ à droite. Les images différence, illustrant le biais en chaque point, sont illustrés à l'aide d'une carte de couleurs en bas.

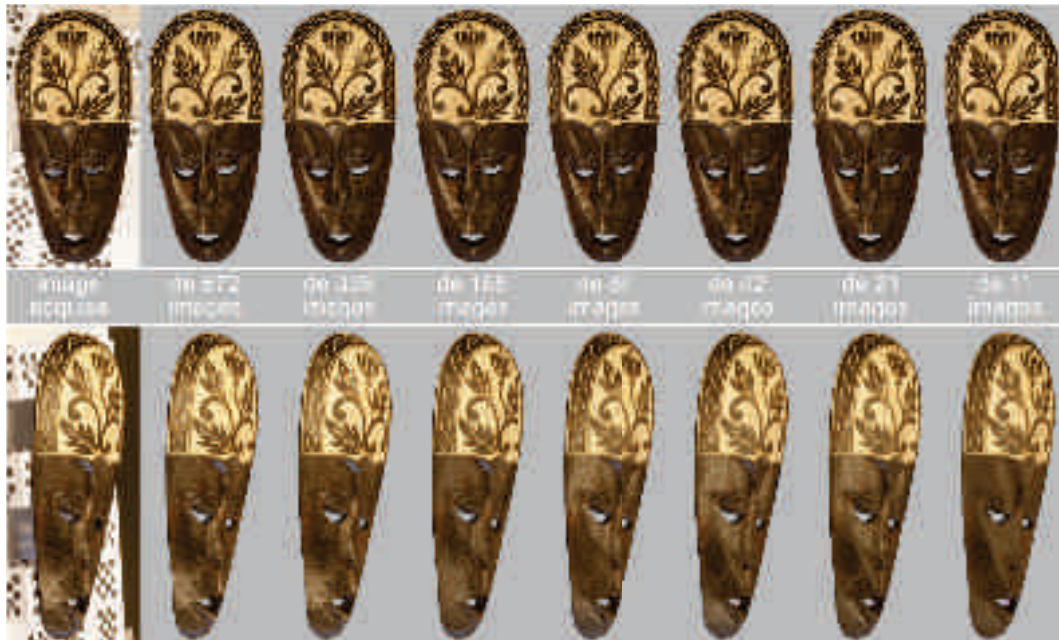


Figure 4.14 : Reconstruction du masque à partir de sous-ensembles des photos acquises. Ces sous-ensembles varient de 672 à 11 images. La photo du haut est présente dans toutes les reconstructions (donc dans les 11 utilisés à droite). La photo du bas n'est utilisée que dans les reconstructions à partir de 672 et 336 photos.

L'énergie E_0 est à écarter car elle génère des fonctions peu souhaitables : le résultat est souvent assombri et l'attache aux données s'en retrouve réduit. Nous l'écartons donc. Les énergies E_1 et E_2 génèrent des résultats stables à un faible coût d'attache aux données, particulièrement l'énergie E_2 . Or, cette énergie souffre d'un manque de réduction d'artefacts en certaines configurations : des couleurs extrapolées peuvent apparaître.

Les extrapolations sont *de facto* fortement réduits sur un domaine fermé (sphérique dans notre cas) si la base est à support global (*i.e.* non compact). Elle est en effet obligée, lorsqu'elle quitte l'hémisphère visible du domaine, de la rejoindre de l'autre côté de l'équateur. Les valeurs du côté visible étant contraintes (par l'attache aux données voire une énergie de stabilisation), ceci implique que la fonction aura une tendance plus faible aux fortes variations par rapport à une base hémisphérique ou à support compact. Ainsi, E_2 est moins sujette aux artefacts avec les harmoniques sphériques qu'avec la base polynomiale (figure 4.12).

Cependant, pour que le problème soit bien contraint, E_2 nécessite 3 échantillons, alors que E_1 produit un résultat déterministe avec un seul. En pratique, ceci n'a pas posé de problèmes flagrants sur les données réelles où un minimum de pixels directionnels étaient présents dans tous les hémisphères : les deux techniques fonctionnent bien dans ce cas. Sur nos données synthétiques, y compris le *Stanford bunny*, nous avons vu que cela est source de nombreux artefacts.

En somme, le choix d'utiliser E_2 pourra se faire dans des conditions particulières : lorsqu'une représentation à domaine fermé est utilisée et lorsqu'un minimum de pixels directionnels par échantillonnage peuvent être garantis. Dans le cas général, E_1 donne des résultats satisfaisants. Nous n'avons pas encore eu à remettre en cause le choix de l'utiliser lors de nos travaux de numérisation.

Une des qualités primordiales de cette reconstruction est illustrée par les figures 4.13 et 4.14 : quelque soit les moyens mis en œuvre (base de fonctions et taille de celle-ci) et les difficultés d'acquisition (échantillonnages), un résultat plausible est rendu. Celui-ci sera tantôt de très forte qualité, tantôt seulement « passable » en fonction de toutes ces conditions externes. Ceci permet de faire des acquisitions plus aisées (à coût réduit) et/ou de générer un résultat correct même pour des objets dont l'acquisition s'avère complexe.

3 Analyse statistique

La gestion du paramètre λ de l'équation (4.6) nous permet de choisir un compromis entre une fonction reconstruite proche des données d'entrée et un résultat stable par rapport aux perturbations recensées. La figure 4.15 illustre l'influence de ce paramètre :

- à $\lambda = 0$, le résultat interpole les données mais exhibe une grande variance générant des artefacts,

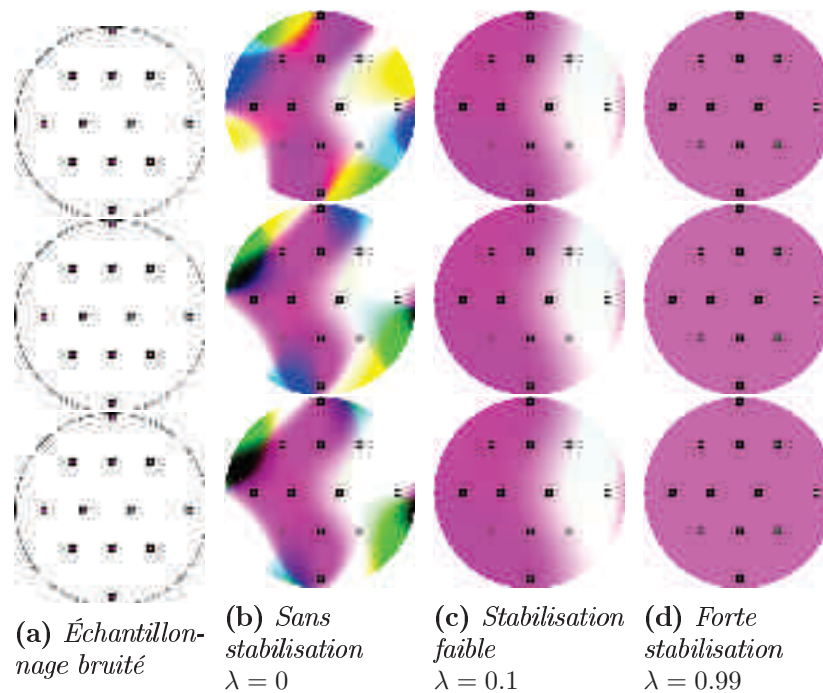


Figure 4.15 : Ajustement de fonctions hémisphériques sur un échantillonnage composé de pixels directionnels bruités à hauteur de 1% dans leurs valeurs colorimétriques (4.15a). Sans stabilisation (4.15b), les fonctions résultantes varient beaucoup en couleur. Avec une faible stabilisation (4.15c), ces effets sont évités. Une stabilisation forte (4.15d) sur-stabilise les fonctions.

- lorsque λ est grand, le résultat est stable et sans artefacts mais ne respecte que peu les données fournies en entrée,
- entre ces deux extrêmes, un compromis que l'on qualifiera d'optimal, réduit la variance tout en générant une bonne attache aux données.

Dans cette section, nous analysons l'évolution en termes de l'attache aux données fournies et de la stabilité par rapport aux variations en fonction de λ . Un outil statistique dûment choisi permettra ensuite de déterminer le paramètre λ générant le compromis optimal entre ces deux entités.

3.1 Biais et variance

A priori nous n'avons à disposition qu'une mesure qualitative permettant de régler empiriquement le paramètre λ et une mesure numérique : la comparaison aux images d'entrée (E_{MSE}) dont la valeur optimale est obtenue lorsque $\lambda = 0$. Cette dernière est souvent la seule mesure utilisée dans la communauté. Elle ne fournit cependant pas d'information sur la robustesse. Un outil automatisé de détermination du compromis optimal serait plus approprié : une mesure de la robustesse nous paraît appropriée. Une évaluation numérique par rapport à une vérité terrain est exclue (cf. introduction

de ce chapitre). Pour cerner les notions de précision et de robustesse, nous allons quantifier leurs inverses : le *biais* introduit et la *variance* du résultat.

Le biais se définit par l'écart entre la fonction reconstruite et les données fournies en entrée, autrement dit l'écart quadratique moyen E_{MSE} . Vu que l'étendue des valeurs est grande, nous utilisons un ratio signal sur bruit [Gersho et Gray, 1992] :

$$PSNR = 10 \log_{10} \left(\frac{E_{max}}{E_{MSE}} \right) \quad (4.10)$$

afin qu'une différence absolue représente un ratio relatif où $10dB$ vaut un facteur 10.

Nous avons déjà évoqué le *biais modèle* qui est le biais introduit *de facto* par une fonction n'ayant pas la capacité de représenter (donc d'interpoler) les données en entrée. À cela va s'ajouter le *biais de stabilisation* qui est dû aux facteurs contre-balançant l'erreur quadratique moyenne : dans notre cas, les énergies de stabilisation.

Définition

Biais de stabilisation : tout ou partie du biais, à savoir la part due à la prise en compte de la stabilisation.

Ce biais augmente avec λ et nous aurons donc tendance à vouloir limiter cet effet, donc à préférer λ petit.

La variance se comprend intuitivement comme la sensibilité de l'algorithme d'ajustement aux perturbations en entrée : bruit, densité et répartition de l'échantillonnage. Cette quantité est mesurable par la différence entre fonctions résultantes ajustées sur des données d'entrée légèrement différentes. Notre but est de réduire la variance du résultat en influant sur l'algorithme d'ajustement. La variance diminue avec λ , nous souhaitons donc augmenter sa valeur suffisamment pour stabiliser l'ajustement.

La section suivante explique comment nous analysons l'évolution de ces entités par rapport à λ .

3.2 Compromis entre qualité et robustesse

Le besoin de stabilisation dépend de nombreux facteurs :

- le bruit induit par l'acquisition et la reconstruction,
- la distribution des pixels directionnels dans l'hémisphère,
- la quantité et les couleurs des pixels directionnels,

- la base de fonctions choisie,
- la taille de cette base,
- l'espace de couleur.

Une façon de déterminer λ localement en fonction de ces perturbations eut été de les modéliser analytiquement. Ceci s'avère infaisable. Certaines de ces propriétés sont certes aisément quantifiables (quantité de pixels directionnels, base de fonctions choisie, taille de la base, espace de couleur) mais tous les facteurs sont très interdépendants. Ainsi, un problème bien contraint par son ratio de nombre d'échantillons sur degrés de liberté de la fonction (*i.e.* taille de la base), devient sur-contraint si les valeurs colorimétriques des pixels directionnels ne sont pas interpolables du fait de la capacité de représentation de la fonction. Ceci n'est qu'un des exemples d'interdépendance de ces facteurs. De plus, certains facteurs ne sont pas quantifiables : à l'échelle macroscopique, un modèle de bruit n'est pas estimable, tellement il y a de bruits, propres au matériel et aux algorithmes utilisés, qui se composent pour venir perturber les valeurs.

Ainsi, plutôt qu'une approche analytique, nous suggérons une approche statistique permettant de proposer une solution au problème de détermination du compromis optimal. Ce compromis se définit pour un échantillonnage donné. Dans la section 3.4, nous expliquons comment mettre en œuvre cette approche statistique sur un maillage entier dans des temps de calcul raisonnables.

Erreur de prédiction \hat{E} . Si l'on traçait le biais et la variance sur un graphe d'abscisse λ , elles se comporteraient de façon parfaitement contraire : le biais croît alors que la variance décroît. L'*erreur de prédiction*, notée \hat{E} , est une mesure statistique qui inclut ces deux entités.

Définition

Erreur de prédiction : écart entre fonction ajustée et de nouveaux échantillons (*i.e.* des échantillons non utilisés pour l'ajustement) issus du même modèle – non observable – de radiance.

Vocabulaire

En statistiques, le vocabulaire suivant s'applique :

Régression : ajustement d'une fonction sur un échantillonnage.

Régularisation : stabilisation d'une régression.

Dans le domaine de l'apprentissage automatique (*machine learning*), le vocabulaire suivant s'applique :

Ensemble d'entraînement : échantillonnage sur lequel une méthode est ajustée.

Ensemble de validation : échantillonnage sur lequel le résultat d'un ajustement est testé (*i.e.* sur lequel la validité du résultat est mesurée).

\hat{E} estime la capacité d'une méthode de régression à générer un résultat représentant bien une radiance donnée et non un échantillonnage particulier de celle-ci. Dans notre contexte, la méthode de régression (régularisée ou non), ajuste une fonction sur un *ensemble d'entraînement* (*i.e.* un échantillonnage de la radiance). \hat{E} se munit (nous expliquons comment dans le paragraphe « Calcul de \hat{E} » à la page suivante) d'un second échantillonnage issu de la même radiance, appelé – *ensemble de validation* – afin de valider le résultat en mesurant l'écart entre la fonction ajustée et cet ensemble. Ainsi, l'erreur de prédiction englobe le biais, vu que tous les pixels directionnels de l'ensemble de validation proviennent du même modèle de radiance (avec tous ses défauts : bruit, répartition, etc.). De même, elle englobe la variance en ce sens qu'une méthode de régression non régularisée génère un résultat fortement variant en fonction de l'ensemble d'entraînement et qui sera mal adapté à l'ensemble de validation.

Graphes. Étant donné une méthode de régression, \hat{E} peut être définie pour tout $\lambda \in [0, 1]^7$. Nous expliquons dans la section suivante comment calculer \hat{E} . Il est alors possible de tracer la courbe de cette erreur (voir figure 4.16). L'erreur est grande à $\lambda = 0$ car la variance est élevée. Lorsque la régularisation diminue la variance (et fait augmenter en moindre mesure le biais), \hat{E} diminue fortement jusqu'à atteindre un minimum. Ensuite, \hat{E} remonte lorsque le gain en biais dépasse la perte de variance. Lorsque λ atteint de grandes valeurs, la courbe est élevée du fait du grand biais introduit par la régularisation, et ce malgré un résultat désormais stable.

Vocabulaire

Compromis : pondération entre attache aux données et stabilisation.

Compromis optimal : compromis qui minimise l'erreur de prédiction.

7. La valeur maximale $\lambda = 1$ n'a pas de sens car il n'y aurait plus aucune contrainte d'attache aux données.

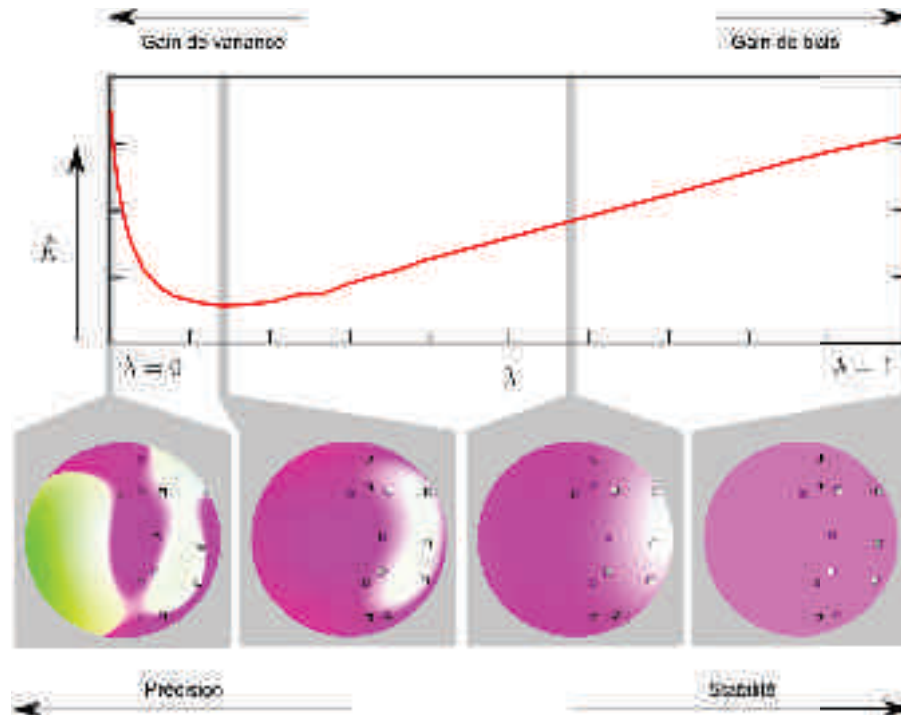


Figure 4.16 : Courbe de \hat{E} en fonction de λ sur un exemple de 12 pixels directionnels mal répartis et une base permettant de les interpoler.

L'erreur \hat{E} minimale correspond ainsi au compromis optimal entre précision et stabilité. En traçant ces graphes sur différents exemples et pour les différents algorithmes (*i.e.* les trois variantes par l'énergie), nous pouvons comparer leurs courbes, notamment afin de définir celle qui aura la plus faible erreur de prédiction minimale. Dans la section 3.3, nous analysons également l'impact de la densité d'échantillonnage et du choix de la base de fonctions sur les courbes.

Calcul de \hat{E} . Le calcul de l'erreur de prédiction est un procédé statistique basé sur des tirages répétitifs d'ensembles d'entraînement et de validation. Nous nous situons dans un cadre où nous n'avons pas d'hypothèse sur la quantité ou la répartition des données, ni même du bruit qu'elles contiennent. Nous savons seulement qu'en certains points surfaciques, elles peuvent être très creuses et mal réparties. Déduire un modèle statistique de ces perturbations est complexe, tout comme en déduire l'erreur de prédiction.

L'observation du modèle de radiance est impossible : nous ne connaissons pas la fonction génératrice de la radiance d'un point de la surface, qui est le résultat d'une interaction lumière-matériau. Nous devons donc exclure les méthodes de calcul de \hat{E} basés sur la génération de données à partir d'une telle fonction. On se basera uniquement sur les échantillons acquis.

De même, il faut exclure les méthodes nécessitant un échantillonnage de grande taille.

La technique de validation croisée découpe l'échantillon en plusieurs sous-ensembles. Une fonction est ajustée sur un ensemble et testée par rapport aux autres (*i.e.* E_{MSE} est évaluée). Une quantité minimale de données est nécessaire pour cela.

La validation croisée *leave-one-out* est une variante qui sépare les échantillons en un ensemble d'un seul pixel directionnel et un ensemble contenant le reste. Ceci se fait de façon aléatoire et répétée. La fonction est ajustée sur le premier ensemble et validé avec le pixel directionnel non utilisé. Cette méthode ne prend pas bien en compte l'amplification de la variation due à la mauvaise répartition dans l'hémisphère : un échantillonnage bien réparti l'est toujours avec un échantillon en moins. Nous l'excluons donc également.

Nous allons en revanche calculer l'erreur de prédiction avec un procédé de *bootstrapping*, celui-ci ne nécessitant pas de fonction génératrice ni même une quantité minimale d'échantillons. Ce procédé permet la génération d'une grande population statistique à partir d'un seul échantillonnage. En entrée, un ensemble T de N pixels directionnels est fourni. Il s'agit alors d'itérativement diviser T en un ensemble d'entraînement (le *bootstrap*) et un ensemble de validation. Les fonctions sont ajustées sur les bootstrap et l'erreur est ensuite calculée par rapport à l'ensemble de validation.

Le procédé requiert trois étapes [Hastie *et al.*, 2001] :

- B bootstraps $\{T_1, \dots, T_B\}$ sont créés. Chacun des T_b est un ensemble composé de N pixels directionnels choisis aléatoirement et avec remise (donc avec doublons) parmi ceux de T .
- Une fonction \hat{R}_b est ajustée sur chaque bootstrap T_b .
- Pour tout pixel directionnel $n \in T$, l'erreur quadratique est moyennée sur tous les bootstraps ne le contenant pas : c'est à dire pour tout R_b ajusté sur un ensemble privé de n . L'erreur est finalement composée de la moyenne de ce calcul sur tous les N pixels directionnels.

$$\hat{E} = \frac{1}{N} \sum_{n=1}^N \frac{1}{\#\{b/n \notin T_b\}} \sum_{b/n \notin T_b} \left(n^n - \hat{R}_b(\omega_r^n) \right)^2 \quad (4.11)$$

L'annexe A donne l'algorithme formel en vue d'implanter cette méthode.

Une propriété intéressante du bootstrapping est que l'on peut augmenter la population statistique (B) indépendamment du nombre N de pixels directionnels en entrée. Ceci nous permettra de faire des tests significatifs sur des données réelles, dont la densité locale d'échantillonnage est incontrôlée. Empiriquement, nous avons choisi $B = 10^4$ bootstraps afin que nos expérimentation donnent deux chiffres significatifs à \hat{E} . Il est à noter que le calcul de 10^4 bootstraps prend en pratique une dizaine de secondes⁸.

8. Pour des harmoniques sphériques à 4 niveaux sur un processeur *Intel Core i5*.

3.3 Résultats quantitatifs

Besoin de stabilisation. Dans un premier temps, il est utile d’observer la corrélation entre l’analyse statistique et les constats qualitatifs de la section (2.3) : l’erreur de prédiction est grande lorsqu’aucune stabilisation n’est faite (cf. toutes les courbes des figures 4.16, 4.17 et 4.18 à $\lambda = 0$). Tous les graphes ont montré la même tendance globale illustrée dans la figure 4.16 : une diminution de l’erreur suivie d’une augmentation.

Sur certains graphes, on observe⁹ que l’erreur de prédiction est plus petite lorsque λ tend vers 1 que lorsque $\lambda = 0$. Ceci souligne le besoin indéniable de stabilisation auquel nous répondons dans ce chapitre : la variance maximale est supérieure au biais maximal, donc l’erreur de prédiction est inférieure lorsqu’on choisit de représenter la couleur par une fonction constante que par une fonction interpolante. Dans ces cas, stabiliser améliore numériquement la « qualité prédictive » des fonctions ajustées.

Les graphes montrent également qu’une faible stabilisation (λ petit) suffit à diminuer l’erreur de prédiction. Cela permet de ne pas introduire trop de biais tout en favorisant l’attache aux données. Par exemple une petite stabilisation ($\lambda = 0.01$) a permis de limiter la réduction de précision (PSNR moyen sur l’objet) de $21.51dB$ à $19.59dB$ dans la figure 4.11d. Sur l’exemple synthétique d’un échantillonnage creux et mal réparti de la figure 4.17b (*i.e.* un cas nécessitant une forte stabilisation), la pondération par $\lambda = 0.05$ est suffisante. Dans le cas de besoin plus faible de stabilisation – échantillonnage dense couvrant l’hémisphère (figure 4.17c) –, la même valeur $\lambda = 0.05$ permet de ne dégrader la précision que de très peu : le PSNR de cet exemple descend de $10dB$ à $9dB$.

Dans la suite nous montrons quelques graphes illustrant le comportement de l’ajustement. Nous avons fait une sélection représentative tout en variant les bases de fonctions. Ces graphes confirment tout ce que nous avons illustré dans les résultats qualitatifs.

Comparaison des énergies. La plupart des graphes, notamment ceux de la figure 4.17, montrent une erreur de prédiction élevée (supérieure à 1) pour $\lambda = 0$. La tendance à la diminution des graphes dans un premier temps (λ petit), permet de constater que les trois énergies stabilisent bien l’ajustement. C’est par leur capacité à atteindre un minimum le plus bas possible – notre principal critère – qu’elles diffèrent, ainsi que par leur capacité à ne pas augmenter le biais rapidement.

Toutes les courbes nous montrent que l’énergie E_0 est moins performante en ce sens : l’erreur de prédiction descend difficilement au même niveau que courbes des deux autres énergies et si elle le fait, remonte rapidement ensuite : le biais introduit par la création de noir devient rapidement pénalisant. Ceci est flagrant sur les graphes des

9. Ceci n’est pas illustré dans nos figures car les graphes sont limités à l’intervalle $\lambda \in [0, 0.5]$ pour des raisons de lisibilité.

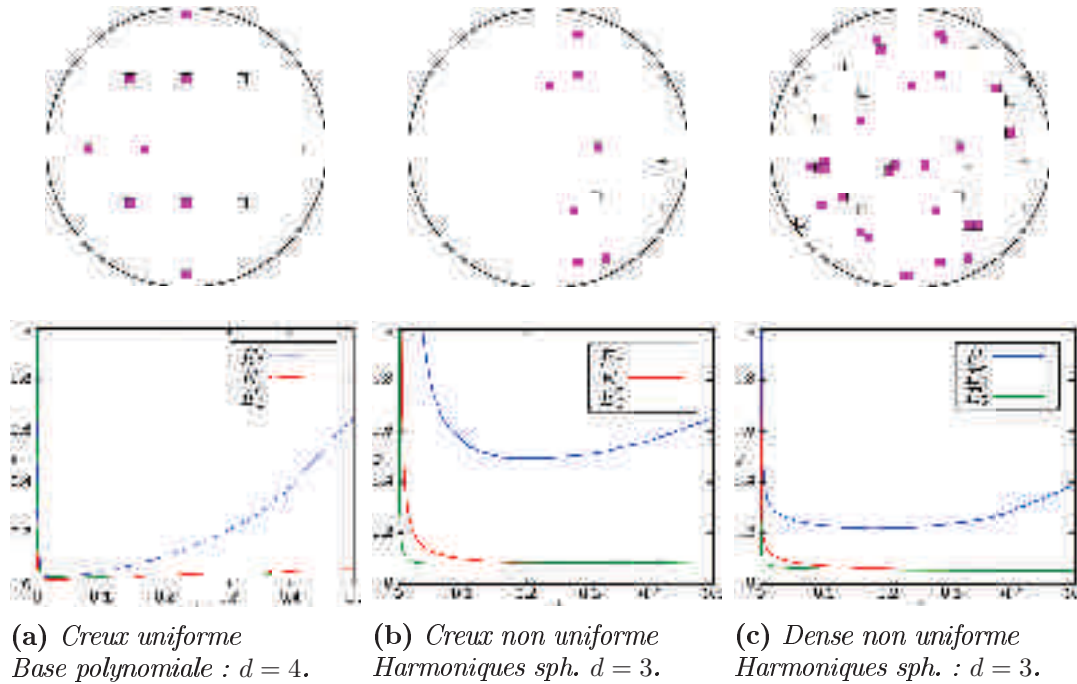


Figure 4.17 : Graphes d'erreurs de prédiction sur divers échantillonnages et pour diverses bases de fonctions.

figures 4.17b et 4.17c. Les graphes confirment également que cette énergie fonctionne mal pour les harmoniques sphériques, en particulier dans la figure 4.18.

Les énergies E_1 et E_2 obtiennent en général de bons résultats numériques : leurs courbes atteignent souvent des minima proches, mais pour d'autres valeurs de λ (plus petit pour E_2). Les graphes des figures 4.17 et 4.18 confirment cela.

3.4 Solutions pratiques

Sur un objet entier, il y a bien évidemment un grand nombre de configurations possibles. Or, nous avons présenté le calcul d'une valeur λ optimale par échantillonnage. Il n'est pas envisageable, de par les temps de calcul, de lancer un processus de bootstrapping sur tous les hémisphères d'un maillage (plus d'un million par objet dans nos exemples). Nous suggérons en revanche qu'une valeur constante peut être définie pour λ . Pour ce faire, nous avons établi un protocole pragmatique.

Nous avons calculé les courbes $\hat{E}(\lambda)$ pour 10 000 points surfaciques choisis aléatoirement (correspondant à 0,05% des sommets du *double dragon* par exemple). En chaque λ nous avons extrait les 95^{ème}, 98^{ème} et 99^{ème} centiles, de telle sorte que 95% (respectivement 98% et 99%) des valeurs des courbes en un point soient inférieurs. Ceci permet de définir les courbes correspondant grossièrement à l'effet de

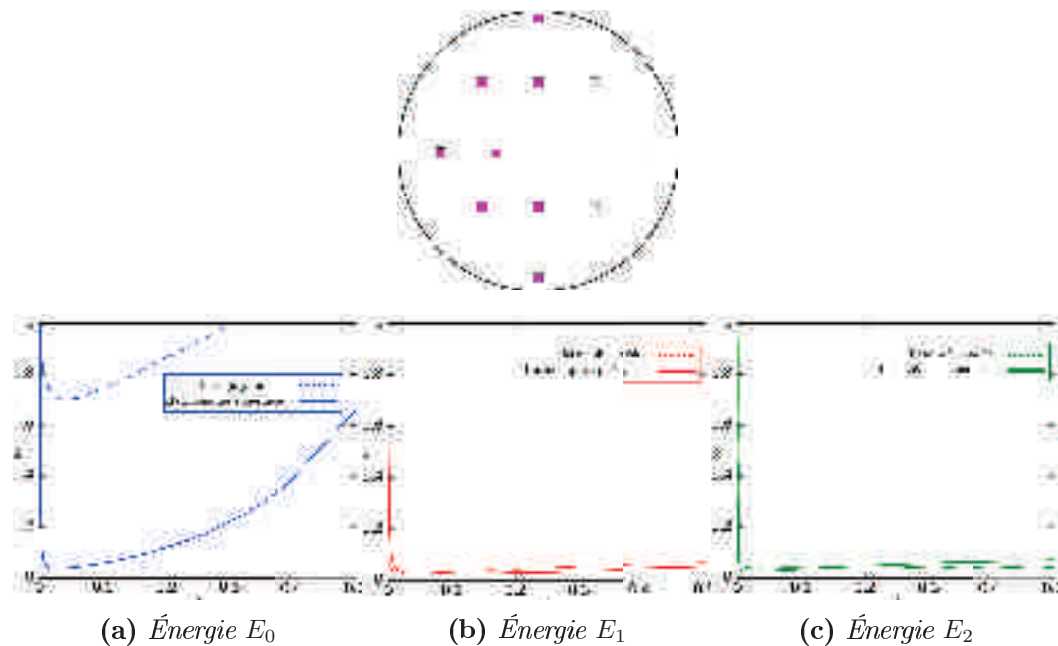


Figure 4.18 : Comparaison entre les effets de la stabilisation des trois énergies en base polynomiale (degré 4, $N = 15$, lignes pointillées) et harmoniques sphériques (degré 3, $N = 16$, lignes pleines) : ajustement sur un échantillonnage creux ($K = 12$) et uniforme. En harmoniques sphériques, E_0 est inefficace, alors que E_1 et E_2 exhibent des résultats similaires dans les deux bases.

stabilisation pour respectivement les 5%, 2% et 1% pires échantillonnages. La figure 4.19 en est le résultat pour l'objet *double dragon*.

L'analyse que nous proposons de ces courbes, et qui permettra de déterminer le λ optimal, est la suivante. La courbe des 95% nous montre qu'une grande majorité de points de la surface ne nécessitent pas de forte stabilisation. Il est ainsi important de garder λ le plus petit possible dans la fourchette de valeurs basses. Les courbes des 98% et 99% représentent la minorité de points mal échantillonnés. Il est également important de choisir λ suffisamment grand pour que leurs courbes descendent proche du minimum. En conclusion, après avoir menée cette procédure pour les trois énergies, nous déduisons que les fourchettes de valeurs satisfaisantes de λ sont les suivantes :

- entre $[0.01, 0.05]$ pour E_0 ,
- entre $[0.01, 0.05]$ pour E_1 ,
- entre $[0.001, 0.005]$ pour E_2 .

Ces valeurs ont été calculées pour un objet acquis avec le même procédé d'acquisition et de reconstruction : celui présenté au chapitre 1. Nous avons constaté qu'elles sont également applicables pour d'autres objets acquis avec notre procédé. Nous ne savons cependant rien de ce qu'il en est pour d'autres procédures d'acquisition, vu que le bruit inhérent aux données y est intimement lié. Il suffira cependant de reproduire

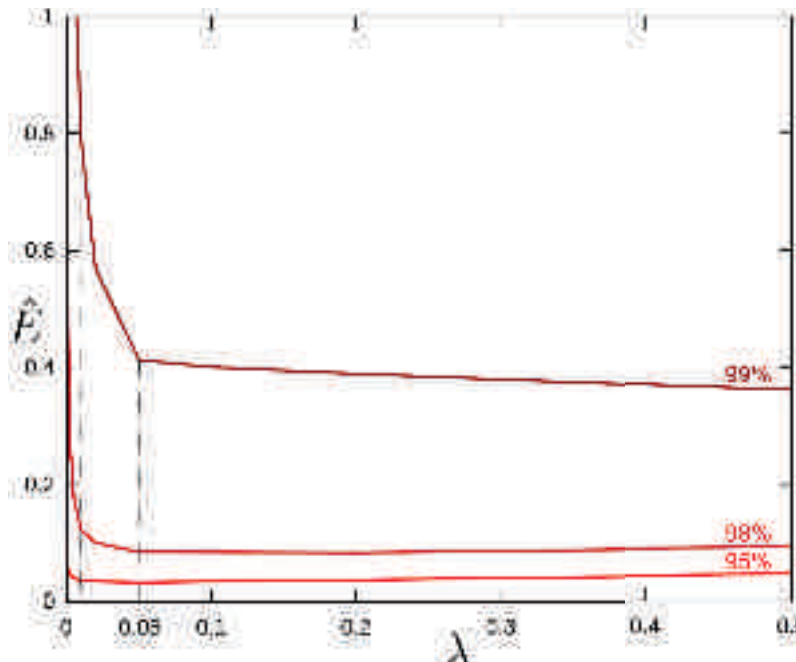


Figure 4.19 : Pourcentages de points nécessitant une forte stabilisation. Les courbes se lisent comme suit : 95% (respectivement 98% et 99%) des échantillons ont une erreur de prédiction inférieure à la courbe correspondante. Stabilisée avec E_1 .

au besoin le protocole de détermination du λ optimal dans un nouveau contexte.

Pour différentes bases et tailles de bases, nous avons constaté des variations mineures, si bien que les mêmes compromis ont pu être utilisés en pratique, comme nous l'avons fait pour le masque reconstruit avec des tailles de base sensiblement différentes dans la figure 4.13. Une base non testée qui nécessiterait un compromis différent ne pose pas de problème, car en règle générale, la base sera choisie une fois pour toute et le procédé de bootstrapping pourra être fait directement avec cette dernière.

4 Conclusion

4.1 Bilan

Dans ces travaux, nous avons pu faire un constat, celui du besoin de robustesse dans le cadre de la reconstruction de champs de lumière surfaciques à partir d'images. Nous y avons répondu en proposant une méthode générique en ce sens qu'on laisse à l'utilisateur le choix :

- de la base de fonctions (nature et taille),
- des outils d'acquisition avec le bruit qui en résulte,
- de sa façon d'acquérir les données (très planifié et détaillé ou plus rapide).

Cette solution permet désormais une acquisition en espace contraint, voire une acquisition plus rapide et à coût matériel et humain réduit.

En somme, cette méthode permet de reconstruire un champ de lumière surfacique stable et de bonne qualité à un faible sur-coût : sa difficulté d'implémentation et sa complexité sont à peine supérieures à programmer une minimisation au sens des moindres carrés. Elle permet surtout de s'abstraire de la fastidieuse étape de planification détaillée des acquisitions ou d'avoir recours à du ré-ou sur-échantillonnage. De plus, la stabilisation s'intègre bien dans tout type d'acquisition : un résultat plausible et correct est retourné lorsque l'échantillonnage est de faible qualité et/ou que l'utilisateur a choisi une base de fonctions de radiance de capacité réduite. Inversement, un résultat de haute qualité est retourné lorsque les conditions sont bonnes. Une fois le compromis optimal entre qualité et robustesse réglé, il n'y a plus à intervenir lors de la reconstruction.

De façon plus détaillée, nous avons mis en avant le besoin de stabilisation et avons analysé et comparé trois variantes répondant toutes au problème de stabilité, mais à des coûts qualitatifs et de généricité de la base de fonctions différents. Nous avons conclu que l'énergie que nous proposons – celle de *Dirichlet* – est la plus appropriée en général, alors que l'énergie *de plaques minces* peut être appropriée dans des conditions bien spécifiques. Finalement, nous avons proposé un protocole permettant de définir le compromis optimal entre qualité et robustesse. Ce protocole peut être appliqué quelque soit la configuration matérielle et logicielle de numérisation.

Outre les exemples visuels parlants sur des données synthétiques et réels, nous avons mis en place une étude statistique adaptée aux contraintes spécifiques de l'acquisition en espace contraint, à savoir l'absence d'une population statistique représentative ni d'un modèle permettant de la générer. Ceci a donné lieu à des résultats quantitatifs agrémentant les résultats visuels de constats numériques. Nous estimons que ce genre d'analyses est souvent manquant dans de nombreux travaux : la seule évaluation visuelle est souvent considérée comme suffisante. Les premiers retours sur ces travaux rapportent un intérêt particulier pour une telle approche. Nous nous réjouissons d'ailleurs de la tendance liée à l'évaluation subjective de la qualité des maillages. De nombreux travaux récents tentent en effet d'établir une corrélation entre évaluation par une métrique donnée et perception par un panel représentatif d'utilisateurs [Lavoué *et al.*, 2006; Herzog *et al.*, 2012; Čadík *et al.*, 2012]. Nous espérons avoir contribué à une utilisation plus large de ce genre d'outils.

4.2 Perspectives

Il y a bien sûr des améliorations possibles à ces travaux, et des pistes de poursuite.

Actuellement, nous choisissons un compromis optimal (une valeur pour λ) de façon globale sur le maillage. Ceci est dû au fait que nous jugeons trop complexe la

détermination d'une valeur optimale en fonction de chaque échantillonnage : leurs paramètres (densité, répartition, bruit, etc.) sont trop inter-dépendants pour pouvoir analytiquement déterminer le compromis approprié. Les temps de calcul pour prendre ces paramètres en compte seraient de plus trop grands pour répéter le bootstrapping.

Cependant, nous pensons qu'il peut être opportun de faire une classification empirique, notamment sur base du constat de la figure 4.19. En effet, une minorité d'échantillonnages nécessitent une stabilisation. Discriminer entre ceux pour lesquels une stabilisation est utile les autres éviterait de stabiliser (donc de maintenir une précision maximale) sur un bon nombre d'échantillons. Ceci n'est cependant pas trivial : il faut déterminer les critères de classification et éviter les effets de seuil lorsque deux hémisphères voisins n'auront pas été classés de la même façon. Il serait ensuite intéressant de déterminer ces classes de façon non empirique.

Le cas échéant, il conviendrait de reprendre notre étude sur le choix de la norme appropriée à considérer pour l'ajustement. La norme L_4 n'a pas été exclue mais il faudrait avant tout vérifier son utilisation en conditions non contrôlées. La stabilisation nous permet désormais de l'envisager.

Un autre problème ayant fait surface lors de numérisations est celui de parties de la surface dépourvues de pixels directionnels (*i.e.* non visibles dans la moindre photo). Dans cette étude, nous n'avons pas traité ce problème car chaque élément a été considéré indépendamment. Une décision radicale est d'interdire la visualisation de telles parties de la surface : après tout, nous n'avons pas réussi à les photographier : pourquoi alors les visualiser ? Cependant, imaginons une application modifiant l'échelle de l'objet et où il serait possible de se « promener » en tout recoin par exemple. Un tel détournement d'objet numérisé (pour intégration dans un environnement autre que celui d'origine) nécessite de pouvoir générer une visualisation plausible à partir de tout point de vue.

L'exploitation de la cohérence spatiale permettrait de diffuser des pixels directionnels selon certains critères. Ce problème est cependant ardu, comme démontré par les premiers travaux entrepris dans notre équipe à ce sujet [Schmitt, 2013].

De façon plus générale, la diffusion de pixels directionnels permettrait d'augmenter l'échantillonnage d'un hémisphère. Des travaux récents traitent ce problème mais avec une représentation globale et pour l'encodage de la réflectance bidirectionnelle [Ruiters *et al.*, 2012].

4.3 Ouverture aux textures polynomiales

Par souci de cohérence, nous n'avons évoqué que l'application de nos travaux sur l'ajustement de fonctions hémisphériques aux champs de lumière surfaciques. Dans l'article [Vanhoey *et al.*, 2013a], nous montrons qu'elle peut être élargie à des applications voisines, tel le ré-éclairage de textures.

Pour illustration, nous avons appliqué notre contribution aux textures polynomiales



Figure 4.20 : De gauche à droite : dôme rigide d'acquisition ; exemple de reconstruction d'une pierre ré-éclairée ; reconstruction et ré-haussement de la texture résultante afin de visualiser les gravures. Illustrations issues de [Malzbender et al., 2001].

(*Polynomial Texture Maps* (PTM)) [Malzbender et al., 2001]. Une source d'éclairage ponctuelle est déplacée autour d'une scène photographiée à partir d'un point fixe. Il est ensuite possible de ré-éclairer l'objet en simulant un rayon de lumière incident de position variable. Une PTM reconstitue alors l'image en fonction de la direction virtuelle d'éclairage ω_i , continue sur \mathcal{H} .

Il y a de nombreuses applications à ce domaine également, tant culturel que pour intégration dans un monde virtuel. Dans l'article présentant les PTM, une application de ré-éclairage de fresques et tombes égyptiennes est montrée (voir figure 4.20). Le ré-éclairage (et le rehaussement possible *a posteriori*) permet notamment de bien visualiser les inscriptions gravées dans la pierre, ce qui est difficile sans cette application.

La problématique ici est analogue aux champs de lumière surfaciques. À partir d'un nombre fini d'acquisitions photographiques (dans lesquels la direction d'éclairage varie), il faut reconstruire une visualisation continue dans l'espace des directions (d'éclairage). La couleur de chaque pixel sera ainsi définie par une fonction de radiance ajustée sur les pixels directionnels acquis. Pour obtenir une reconstruction robuste de ces données, un échantillonnage dense et quasi-régulier est également demandé, d'où l'utilisation d'un dôme de placement des sources lumineuses (figure 4.20). Même si les problèmes liés à l'échantillonnage sont moindres qu'avec les champs de lumière surfaciques – il n'y a jamais d'occultations car on ne visualise la scène qu'à partir d'un seul point de vue et tous les hémisphères sont orientés exactement dans la même direction – l'estimation de la position de la source lumineuse peut être délicate, notamment lors de la numérisation d'objets fragiles. Ainsi, il se peut qu'il soit impossible de placer une source lumineuse en un nombre d'endroits.

Ceci est mis en exergue sur des données numérisées par nos collègues du CNR de *Pise en Italie*. 36 photos (1000×750 pixels) du *Capitello* (voir figure 4.21) ont été acquises

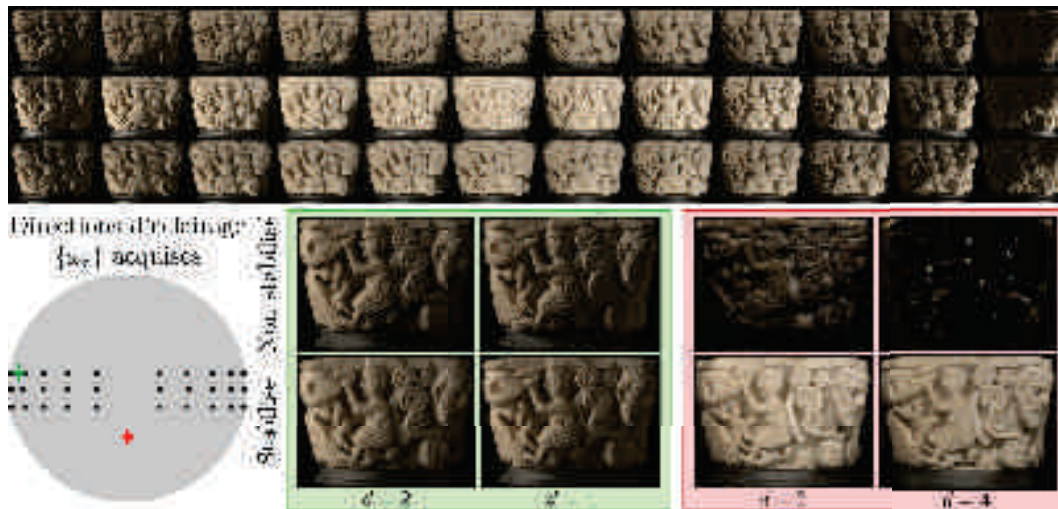


Figure 4.21 : Ré-éclairage du « Capitello » : 36 images acquises sont utilisées pour reconstruire de façon non robuste et robuste un modèle ré-éclairable. Les directions d’acquisition sont illustrées dans le disque et les points de vue reconstruits à droite (vert et rouge) y sont également représentés. La stabilisation de l’ajustement de fonctions apporte ici aussi la nécessaire robustesse et permet de générer un résultat souhaité et attendu.

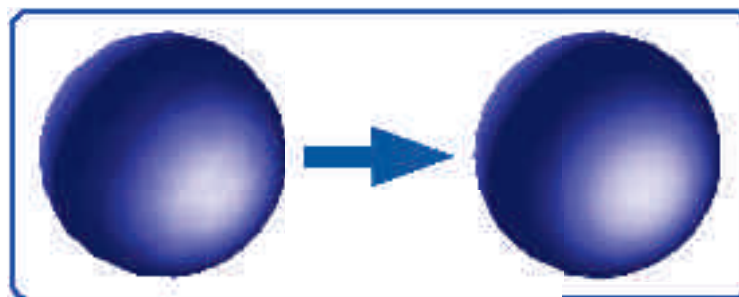
sous des directions d’illumination différentes. Une interpolation non contrainte résulte en de nombreux artefacts (voir « non stabilisé » en bas de la figure 4.21). Grâce à l’ajout de notre stabilisation, ces données sont correctement reconstruites sans recours à un échantillonnage extrêmement dense.

Notre contribution est directement transposable à ces données. Même si le besoin de stabilisation est moins marqué, elle permet aisément de reconstruire une PTM à la fois robuste et avec une bonne attache aux données. Il est à noter que nous avons utilisé les mêmes fourchettes de valeurs de λ pour déterminer le compromis entre qualité et robustesse, bien que les variations en entrée soient moindres (le bruit est réduit car il y a moins d’étapes d’acquisition et de reconstruction) et que l’échantillonnage soit moins creux que dans le pire cas des champs de lumière. Les résultats sont corrects mais une application du protocole de bootstrapping sur ces données pourrait sans doute permettre de gagner en qualité.

Chapitre 5

Visualisation de fonctions de radiance sur la surface

Donner au discret l'illusion du continu.



RÉSUMÉ:

Ce chapitre explique comment interpoler les fonctions de radiance sur le maillage, étape nécessaire pour rendre une couleur sur la surface complète. Dans un premier temps (section 1), nous présentons précisément comment les données sont structurées et interprétées. Nous y abordons notamment les deux façons d'attacher les données au maillage : attaché aux sommets ou dans des texels. Les travaux de ce chapitre s'appliquent à de la radiance attachée aux sommets. Dans la section 2, nous faisons un état de l'art des modes de rendu de maillages sans attribut ou avec des attributs de couleur. Le pipeline GPU et les modes d'interpolation de simples couleurs y sont expliqués. Ceci nous permet ensuite d'expliquer comment rendre un maillage avec attribut de radiance dans la section 3. Pour cela, nous allons interpoler des fonctions symétrisées autour de leur normale afin de rendre les reflets directionnels de façon cohérente. Nous expliquons, notamment à l'aide de pseudo-code, comment programmer cela dans les *shaders* de la carte graphique. Finalement, nous faisons un bilan dans la section 4 et discutons de l'applicabilité de nos travaux aux données de radiance stockées dans des textures.

Sommaire

1	Stockage et interprétation de la radiance	113
2	Rendu d'un maillage	116
2.1	Visualisation sur GPU	116
2.2	Rendu de la couleur simulée	118
2.3	Rendu de couleurs stockées	121
3	Rendu de radiance	122
3.1	Interpolation classique	122
3.2	Interpolation de radiance « symétrisée »	125
3.3	Implémentation	126
3.3.1	Solutions techniques	127
3.3.2	Détails et commentaires	129
3.4	Résultats	130
4	Conclusion	133
4.1	Bilan	133
4.2	Perspective : radiance par texel	134

À l'issue de l'ajustement de fonctions présenté au chapitre précédent, nous disposons de données prêtes à être rendues. Il s'agit d'un maillage surfacique sur lequel des informations de radiance sont réparties, sous forme de fonctions exprimées dans une base. Ces fonctions peuvent être évaluées pour un point de vue donné afin de déterminer la couleur en leurs points d'attache. Dans ce chapitre, nous proposons une nouvelle façon de rendre une couleur sur *toute la surface* par interpolation de fonctions de radiance. Ce rendu donne une meilleure impression de continuité, en particulier lorsque la densité spatiale est faible.

1 Stockage et interprétation de la radiance

Nous présentons ici la structure de nos données dont la figure 5.1 est une représentation schématique. Le modèle tri-dimensionnel que nous manipulons est défini par un maillage surfacique composé de sa partie *topologique* et de ses *attributs*.

La topologie définit la *connectivité* entre *cellules* : dans notre cas il s'agit d'un ensemble de *faces triangulaires* se rejoignant en des *sommets* reliés entre eux par des *arêtes*. Nous considérons par la suite que nous manipulons des variétés. Dans ce cas, une arête est adjacente à deux sommets et deux triangles, un triangle à trois arêtes et trois sommets, un sommet à un nombre quelconque d'arêtes et de triangles. Tous les modèles que nous avons reconstruits rentrent dans ce cadre (et sont par ailleurs des surfaces fermées).

Les attributs sont en général attachés à un sommet car la carte graphique n'accepte que cela en entrée¹. Parmi eux, il y a *a minima* un plongement géométrique dans l'espace. Par ailleurs, il est courant en informatique graphique d'y associer des normales ou des couleurs.

Il existe des théories élégantes et efficaces à implanter pour représenter les maillages avec attributs. Nous utilisons une implantation des cartes combinatoires via la librairie *CGoGN*². Cette théorie et son implantation sont décrites dans l'article [Kraemer *et al.*, 2013]. Alors que la facilité de parcours d'un maillage qu'offre ce modèle est profitable pour l'implémentation de la simplification (chapitre 6), il suffit pour comprendre la problématique de ce chapitre, de raisonner triangle par triangle.

Couleurs par sommet ou par texel. Les attributs liés à la colorimétrie (couleurs, normales, radiance) peuvent être attachés aux sommets ou stockés dans une texture couvrant la surface de l'objet. Dans ce dernier cas, des coordonnées de texture sont associées à tout sommet, afin d'établir un lien entre surface et les informations stockées dans les texels. Nous stockons des fonctions de radiance et la figure 5.1 illustre les deux variantes d'attaches : aux sommets (turquoise) ou aux texels (bleu).

Les deux ont leurs avantages et inconvénients.

1. Nous proposons une ouverture à ce sujet dans le chapitre 7.

2. *Combinatorial and Geometric modeling with Generic N-dimensional Maps* : plateforme développée dans l'équipe IGG, et à laquelle nous avons pu contribuer.

L'avantage de déporter des attributs colorimétriques dans une texture est le découplage entre la résolution du maillage et la résolution de l'information de colorimétrie. Ainsi, un maillage grossier peut être défini tout en maintenant une résolution spatiale fine pour la radiance par exemple. Ceci pose cependant divers problèmes techniques que nous discuterons dans le chapitre 6. De plus, il n'est pas évident qu'une telle décorrélation soit souhaitable, tant la radiance et la géométrie locale sont intimement liés. Ceci est sans doute particulièrement vrai dans le domaine de la numérisation du patrimoine, où l'exactitude des données est importante.

Finalement, la définition d'une texture couvrant le maillage génère une difficulté supplémentaire : celle du paramétrage de la texture sur la surface. Alors que des solutions empiriques ou manuelles existent, les discontinuités aux bords de la texture (*i.e.* aux coutures) posent toujours des problèmes.

Nous avons choisi dans un premier temps d'attacher la radiance aux sommets (les images du masque africain du chapitre 4 ont été générés dans un second temps avec de la radiance dans une texture). Les explications de ce chapitre (ainsi que du chapitre 6 traitant de la simplification) se limitent à ce type de stockage. Dans les conclusions de ces deux chapitres, notamment la section 4, nous discutons de ce qu'il reste à faire pour étendre nos travaux à de la radiance stockée dans une texture.

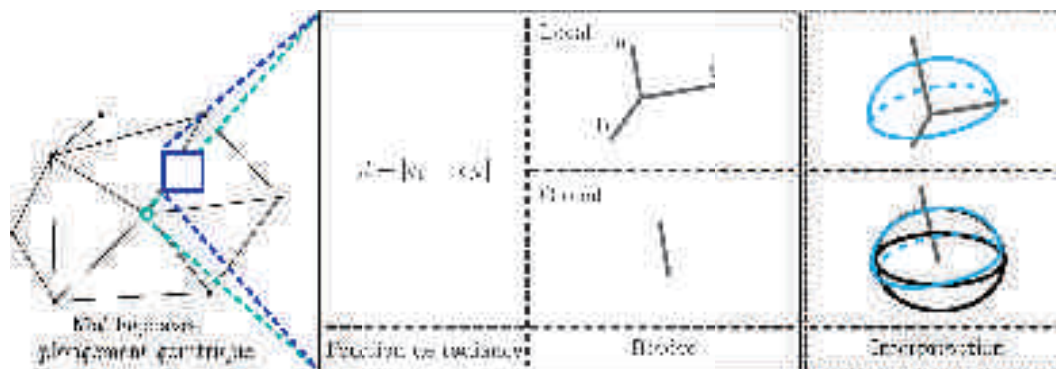


Figure 5.1 : Structure de données du champ de lumière surfacique avec fonctions de radiance locales. La radiance est attachée au maillage surfacique plongé dans \mathbb{R}^3 (à gauche) en un sommet (turquoise) ou un texel d'une texture (bleu). Elle est déterminée par une fonction de radiance (*i.e.* un vecteur de coefficients s'exprimant dans une base implicite) ainsi qu'une information d'orientation (normale) voire de paramétrage (repère de Frenet) selon que le repère est défini localement ou de façon globale respectivement. La base de fonction est paramétrée sur ce repère. Dans le cas local, le repère de Frenet permet d'orienter l'hémisphère visible formant le domaine de définition ainsi que de le paramétrer. Dans le cas global, les coefficients s'expriment sur un domaine sphérique paramétré de façon implicite. Une normale permet alors de déterminer l'hémisphère visible (*i.e.* la moitié « significative » du domaine sphérique).

Repère local ou global. Dans le carré solide de la figure 5.1, nous illustrons comment la radiance est représentée. Il s'agit de stocker la fonction à proprement parler (*i.e.* ses coefficients), ce qui permet de l'évaluer à l'aide de la base implicite que l'utilisateur aura choisie. Cette base peut être paramétrée de deux façons : sur un domaine défini localement ou sur une sphère globale.

Dans le cas local, une normale permet d'orienter l'hémisphère visible sur lequel la fonction de radiance est définie. L'ajout de deux vecteurs paramétrant le plan tangent (tangente et binormale) permet de le paramétrer. Ainsi, le repère de *Frenet* (composé des trois vecteurs) qui a été utilisé lors de l'ajustement de la fonction de radiance doit être fourni en vue d'évaluer la fonction.

Dans le cas global, les coefficients s'expriment sur un domaine sphérique implicite. L'ajustement de fonction aura donc généré une fonction définissant la radiance sur toute la sphère et évaluable pour toute direction de radiance. Il suffit alors de connaître la normale afin de déterminer l'hémisphère visible représentant la partie « pertinente » du domaine sphérique (en bleu clair à droite sur la figure 5.1).

Chaque représentation a ses avantages et inconvénients.

D'un point de vue pratique, le repère local nécessite le stockage de quelques vecteurs : le repère de *Frenet*. Cependant, à nombre de coefficients égal, la fonction exprimée en un repère global a une capacité de représentation inférieure : la moitié de son domaine n'est jamais utilisé, et il faut ainsi stocker davantage de coefficients pour obtenir une qualité équivalente. Au chapitre 3, nous avons présenté différentes bases de fonctions, qui peuvent tantôt se définir dans un repère quelconque, tantôt dans un repère local uniquement. Ceci peut également guider le choix de l'utilisateur.

Du point de vue mathématique, le calcul sur les fonctions exprimés dans des repères locaux est plus complexe : contrairement à un repère global, il faut avoir recours à des rotations dans l'espace pour comparer deux fonctions ou les interpoler par exemple. Un domaine global nécessite cependant de déterminer une fonction sur toute la sphère, ce qui en soi est un non-sens. La moitié « non pertinente » n'a forcément pas été acquise, et il n'est alors pas évident d'y déterminer une valeur de fonction. Un comportement par défaut (*i.e.* y mettre du noir) crée une discontinuité qui compliquera les calculs d'interpolation ou de distance entre fonctions, nécessaires dans nos travaux. Toute autre solution paraît peu intuitive.

Comme pour la base ou l'espace de couleur, nous ne faisons pas de choix privilégié quant au repère. L'utilisateur peut avoir de bonnes raisons de choisir l'un ou l'autre. À titre d'exemple, dans [Wood *et al.*, 2000], un repère global est choisi afin de pouvoir facilement exploiter la cohérence entre fonctions de radiance dans un objectif de compression par quantification. Inversement, c'est la simplicité de la base qui a motivé certains travaux à utiliser une représentation locale [Malzbender *et al.*, 2001; Larue et Dischler, 2006] et le grand ratio capacité de représentation sur nombre de coefficients qui en a motivé d'autres [Gautron *et al.*, 2004].

Les travaux que nous présentons dans ce chapitre portent sur l'interpolation de fonctions de radiance. La mise en œuvre est plus ou moins complexe selon le choix du repère (local ou global). Nous nous efforçons donc à proposer des solutions fonction-

nant dans les deux cas et discutons l'extensibilité de nos contributions aux textures dans la section 4.

2 Rendu d'un maillage

Nous présentons le pipeline GPU en section 2.1. Il traite des données par sommet exclusivement, ce qui implique qu'il faut inévitablement définir une interpolation afin d'obtenir des informations en tout point de la surface. Dans les sections 2.2 et 2.3, nous expliquons comment ceci est mis en œuvre dans le cas de couleurs stockées ou calculées (par le biais d'un modèle d'éclairage) par sommet.

2.1 Visualisation sur GPU

Le schéma de la figure 5.2 représente le pipeline GPU et sert d'illustration à ce chapitre. Il ne représente que les parties d'intérêt pour nous (*i.e.* les parties programmables) et n'est en ce sens pas exhaustif. Ces parties, appelées *shaders*, sont illustrées sur un exemple de traitement d'un maillage muni d'une topologie, d'un plongement géométrique et d'un attribut de couleur. Les phases de calcul en gris sont faites automatiquement par le GPU, alors que les parties pouvant s'exécuter en parallèle (les trois *shaders*) sont programmables.

Définition

Shader : entité programmable de la carte graphique. Le langage de programmation appliqué est le *GLSL*. Ces petits programmes traitent de composants unitaires (sommet, polygone ou pixel) de façon massivement parallèle.

Vertex shader : l'entité programmable traitant individuellement chaque sommet du maillage.

Geometry shader : l'entité programmable traitant individuellement chaque polygone du maillage, dans notre cas des triangles.

Fragment shader : l'entité programmable traitant individuellement chaque pixel écran. La surface couverte par un pixel est appelée un *fragment*.

1—Traitement des sommets. Le GPU prends en entrée une liste de sommets avec leurs attributs et l'information de connectivité triangulaire applique un traitement à chacun grâce à l'exécution du *vertex shader* (figure 5.2 en haut). Celui-ci traite parallèlement et indépendamment tous les sommets. Son rôle est d'opérer les transformations géométriques permettant de passer de coordonnées dans un repère « monde » aux coordonnées en repère écran. Ce dernier dépend des transformations

faites par l'utilisateur lors de la navigation virtuelle dans la scène : rotations et translations, comprenant des mises à l'échelle. D'autres attributs peuvent également être traités ici.

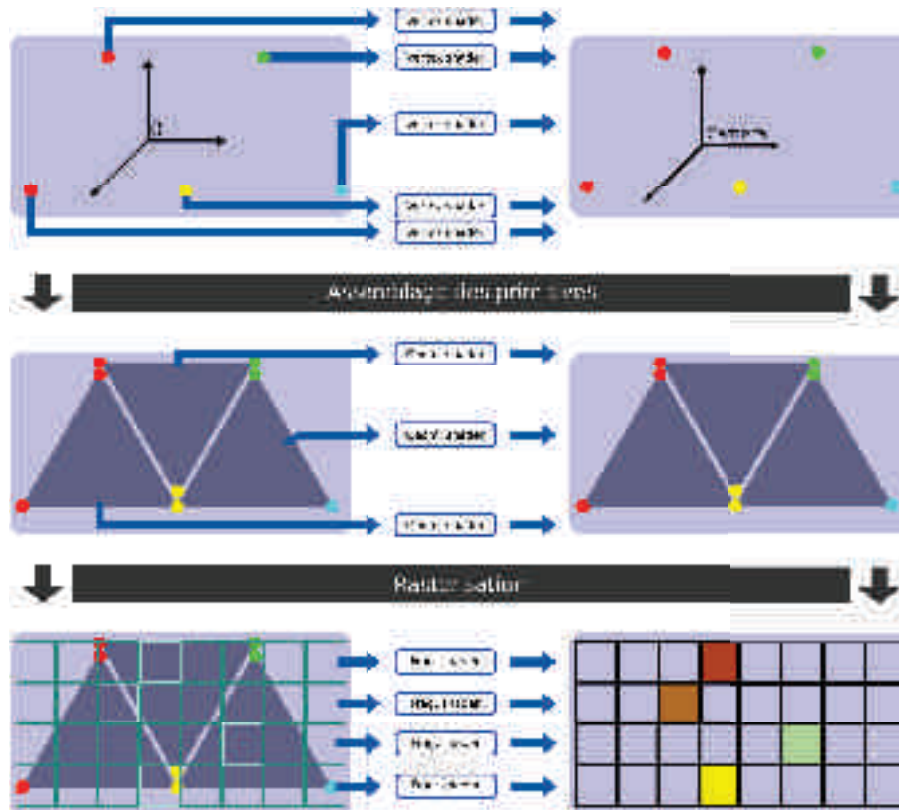


Figure 5.2 : Pipeline de calcul sur le GPU : étapes programmables. Illustration à l'aide d'attributs de géométrie et de couleur par sommet.

2—Traitement des faces. Suite à cela sont assemblées les *primitives géométriques*, dans notre cas les triangles : la connectivité entre sommets est ajoutée et permet de traiter parallèlement les faces grâce à l'exécution du *geometry shader* (figure 5.2 au centre). Son rôle est de modifier la géométrie, par exemple en modifiant les coordonnées des sommets d'un triangle en les déplaçant le long de la normale pour faire du *height mapping* ou en subdivisant le triangle par l'ajout d'un sommet au milieu de ce dernier.

Le schéma illustre le comportement par défaut : les attributs de chaque sommet sont simplement transmis à l'étape suivante³.

Notons que chaque triangle accède aux trois sommets qui lui sont adjacents : il

3. Lorsque l'on dit « par défaut », il s'agit d'une part du comportement historique du GPU, avant que la possibilité ne soit laissée au programmeur de gérer le code de ces étapes, d'autre part du programme minimal que nous aurons programmé, sauf lorsque nous stipulons le contraire.

peut modifier les valeurs de ses attributs. Par souci de flexibilité et pour éviter les conflits, chaque sommet est ainsi considéré une fois par triangle, d'où la duplication des sommets dans notre schéma. Nous expliquons à la fin de ce chapitre comment exploiter ce *shader* pour astucieusement transmettre nos attributs jusqu'à la fin du pipeline pour l'interpolation de fonctions de radiance.

3–Traitement des fragments. Finalement, la *rasterisation* découpe la scène à visualiser selon les pixels qui sont affichés à l'écran afin de traiter parallèlement ces derniers grâce à l'exécution du *pixel* ou *fragment shader* (figure 5.2 en bas). Son rôle est d'agrémenter chaque fragment d'une couleur (dans l'espace *RVB*), en vue de l'affichage à l'écran.

En entrée de ce *shader*, les attributs rattachés aux trois sommets d'un triangle sont interpolés bi-linéairement (selon la coordonnée barycentrique du fragment courant dans le triangle). Ainsi, la position géométrique (mais également la couleur ou d'autres attributs) est connue pour chaque pixel.

2.2 Rendu de la couleur simulée

Dans le cas d'un maillage sans attribut de colorimétrie, la couleur est souvent simulée par l'application d'un modèle de réflectance et d'un éclairage pré-défini. Lors du rendu, le point de vue est connu, ce qui permet de calculer la radiance dans la direction d'observation. Contrairement à ce que nous avons l'habitude de représenter dans cette thèse, cette radiance simulée n'est pas stockée, mais calculée à la volée. Seul le modèle de réflectance et l'éclairage sont pré-définis. Nous décrivons ici l'application d'un modèle de *Blinn-Phong* sur le maillage entier. Pour rappel, ce modèle de réflexion définit la couleur d'un point p grâce à l'équation

$$f_{BF}(\omega_i, \omega_r) = k_a I_a + k_d \langle \omega_i, \mathbf{n} \rangle I_d + k_s \langle \omega_r, \mathbf{r} \rangle^\alpha I_s$$

En plus des paramètres globaux et connus (k_a , k_d , k_s , I_a , I_d et I_s), ce calcul nécessite les paramètres locaux suivants :

- le vecteur ω_i entre p et la source lumineuse⁴,
- le vecteur ω_r entre p et le point de vue,
- le vecteur normal \mathbf{n} à la surface en p .

Vocabulaire

Les éléments suivants sont des données globales accessibles dans tout *shader* des pseudo-codes à venir.

vec3 posCam : la position de la caméra.

vec3 posLum : la position de la source lumineuse.

transfo : la fonction de transformation géométrique.

BF : la fonction régissant le modèle de *Blinn-Phong*.

4. Nous ne considérerons qu'une seule source lumineuse afin de faciliter l'explication.

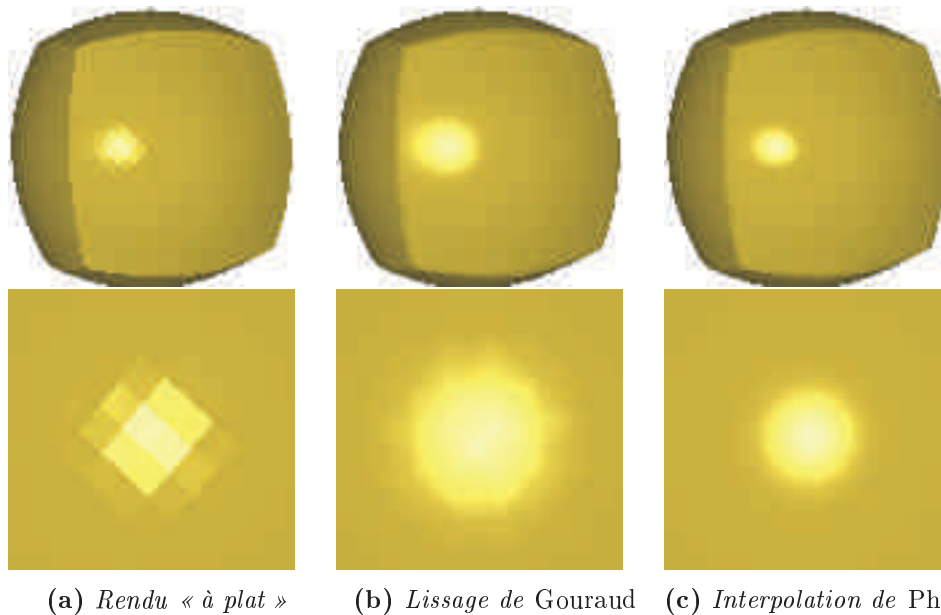


Figure 5.3 : Rendus de surfaces avec couleurs simulées par modèle de Blinn-Phong.

Étant donnés une fonction de réflectance globale et une géométrie par sommet, il y a trois façons de calculer une couleur par fragment :

1–Rendu « à plat ». En chaque triangle, la normale est déterminée. En chaque fragment, la fonction de réflectance peut être évaluée car \mathbf{n} , ω_i et ω_r y sont déterminés. Ainsi, la *geometry shader* calcule la normale de la face, la transmet au *fragment shader* qui détermine les directions ω_i et ω_r en un fragment afin de calculer la couleur rendue. L'algorithme (1) décrit le calcul de chaque *shader* sous forme de pseudo-code.

Algorithme 1 Rendu à plat.

<pre>// Vertex shader IN // par sommet Position p; CALCUL vec3 pt = transfo(p); OUT Position = pt;</pre>	<pre>// Geometry shader IN // par triangle Positions p[3]; CALCUL vec3 n = (p[1]-p[0]) ^ (p[2]-p[1]); OUT Positions[3] = p; Normales[3] = {n,n,n};</pre>	<pre>// Fragment shader IN // par fragment Position p; // interp Normale n; CALCUL vec3 Wr = posCam - p; vec3 Wi = posLum - p; vec3 c = BF(n,Wr,Wi); OUT FRAG_POS = p; FRAG_COLOR = c;</pre>
--	--	--

La figure 5.3a montre un exemple de résultat : la couleur est constante pour tout fragment d'une face⁵. On appelle ceci le *flat shading*, d'après les triangles

5. *N.B.* : ceci n'est vrai qu'à condition qu'observateur et source lumineuse soient suffisamment

dont on voit qu'ils sont plats.

2–Lissage de *Gouraud*. Le lissage de *Gouraud* [1971] propose de déterminer une normale variant de façon continue sur le triangle. Pour ce faire, les normales sont pré-calculées par sommet en moyennant celles des triangles adjacents. Elles sont alors passées au GPU comme attributs rattachés aux sommets et la fonction de réflectance est évaluée dans le *vertex shader* afin de générer une couleur par sommet. Celle-ci est finalement interpolée bi-linéairement dans la phase de rasterisation, afin d'obtenir une couleur sur toute la surface. L'algorithme (2) décrit les *shaders* du lissage de *Gouraud*.

Algorithme 2 Lissage de *Gouraud*.

<pre>// Vertex shader IN // par sommet Position p; Normale n; CALCUL vec3 pt = transfo(p); vec3 Wr = posCam - p; vec3 Wi = posLum - p; vec3 c = BF(n,Wr,Wi); OUT Position = pt; Couleur = c;</pre>	<pre>// Geometry shader IN // par triangle Positions p[3]; Couleurs c[3]; CALCUL OUT Positions[3] = p; Couleurs[3] = c;</pre>	<pre>// Fragment shader IN // par fragment Position p; // interp Couleur c; // interp CALCUL OUT FRAG_POS = p; FRAG_COLOR = c;</pre>
--	---	--

La figure 5.3b montre que le résultat est nettement plus lisse et donc plus réaliste. Néanmoins, cette façon de faire est source d'artefacts lorsque deux triangles voisins sont grands et ont des normales relativement divergentes : on y devine la structure triangulaire sous-jacente.

3–Interpolation de *Phong*. Finalement, *Phong* [1975] propose de maintenir ces normales lisses, mais d'évaluer la fonction par fragment afin d'éviter les étirements de pics spéculaires dont le lissage de *Gouraud* est victime. Ainsi, les normales sont aussi pré-calculées par sommet puis passées comme attribut, mais la couleur n'est pas évaluée par sommet (*i.e.* dans le *vertex shader*) : la normale est interpolée bi-linéairement sur le triangle lors de l'étape de rasterisation et varie ainsi de façon lisse sur la surface. Finalement, le *fragment shader* évalue l'équation de réflectance en chaque fragment afin d'en déterminer la couleur⁶. L'algorithme (3) montre le pseudo-code des *shaders* pour l'interpolation de *Phong*.

La figure 5.3c montre que l'écueil des algorithmes précédents sont évités : l'illusion de continuité est quasi parfaite. Un écueil est toutefois inévitable :

éloignés de la surface afin que la variation des vecteurs lumière-surface et caméra-surface d'un fragment à l'autre soit négligeable.

6. *N.B.* : cette interpolation est une approximation car il serait plus judicieux d'avoir recours à une interpolation sphérique.

Algorithme 3 Interpolation de *Phong*.

<pre>// Vertex shader IN // par sommet Position p; Normale n; CALCUL vec3 pt = transfo(p); OUT Position = pt; Normale = n;</pre>	<pre>// Geometry shader IN // par triangle Positions p[3]; Normales n[3]; CALCUL OUT Positions[3] = p; Normales[3] = n;</pre>	<pre>// Fragment shader IN // par fragment Position p; // interp Normale n; // interp CALCUL vec3 Wr = posCam - p; vec3 Wi = posLum - p; vec3 c = BF(n,Wr,Wi); OUT FRAG_POS = p; FRAG_COLOR = c;</pre>
--	---	--

lorsque des normales de sommets voisins sont fortement divergentes, des artefacts peuvent apparaître. Les figures 5.3b et 5.3c illustrent ces artefacts sur les arêtes vives (effet en « dent de scie »). Elles sont dues au mauvais fondement d'une normale par sommet : le rendu interpolé est différent lorsqu'un triangle adjacent à une arête vive partage un seul ou inversement deux de ses sommets avec cette arête.

En somme, *le rendu à plat* est le mieux défini mathématiquement : il exploite la définition naturelle d'une normale (*i.e.* perpendiculaire au plan déterminé par un triangle). Comme le résume bien l'article de Boubekur et Alexa [2008], l'interpolation de la normale permet de cacher les artefacts liés à la non continuité d'ordre supérieur de la surface, ce qui permet de donner l'illusion d'une surface lisse. C'est ce que font le lissage de *Gouraud* et l'interpolation de *Phong*. Ce dernier représente cependant mieux les effets induits par la lumière sur les fragments : alors que le lissage de *Gouraud* ne calcule la radiance correctement que sur les sommets, l'interpolation de *Phong* la calcule correctement sur tout fragment.

2.3 Rendu de couleurs stockées

En modélisation ou numérisation, il est également possible de pré-définir la couleur : soit en « peignant » virtuellement l'objet, soit en reconstruisant une couleur à partir d'images acquises. Le résultat est généralement une surface *Lambertienne* totalement diffuse : les effets liés à la réflexion ne sont pas encodés. Il y a deux façons de stocker ces données sur le maillage : par sommet ou par texel. Dans les deux cas, une interpolation bi-linéaire des couleurs est souhaitable :

- Si la couleur est stockée par sommet, la couleur d'un fragment situé dans un triangle résulte de l'interpolation des couleurs aux sommets adjacents (voir figure 5.4a). N'ayant pas d'information de couleur par face, il n'est pas possible d'en faire un rendu « à plat » (*i.e.* constant par face).
- Si la couleur est stockée par texel, une coordonnée de texture est associée à

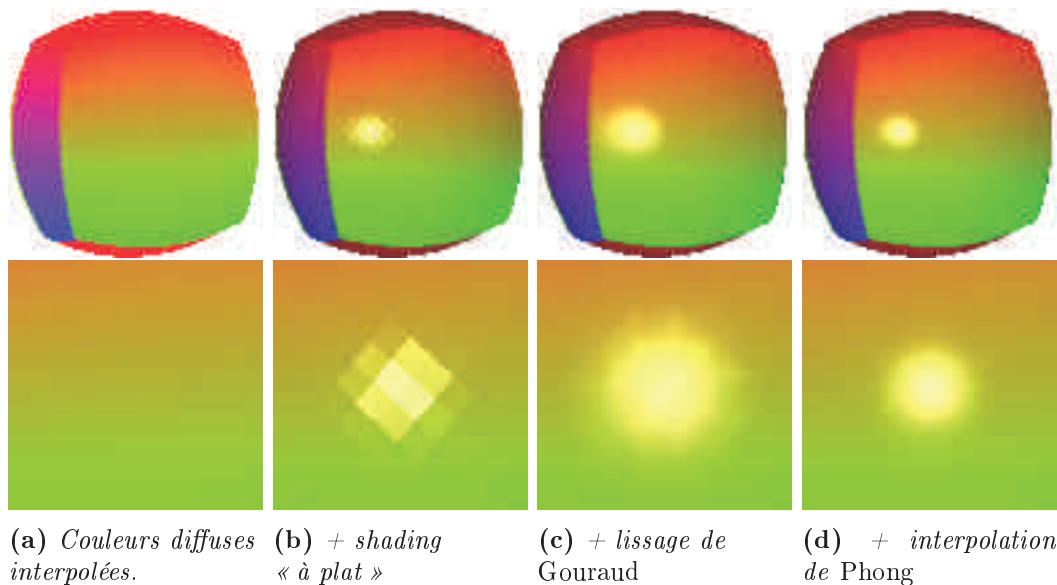


Figure 5.4 : Rendus de couleurs par sommet.

chaque sommet puis interpolée en chaque fragment. La coordonnée qui en découle désigne alors un point dans la texture stockée sur la carte graphique qui se situe entre quatre texels. Ces quatre texels sont alors également interpolés (bi-linéairement) afin de déterminer la couleur résultante.

La figure 5.4a illustre le résultat obtenu.

Dans le cas de modèles « points », il est d'usage d'ajouter la partie spéculaire du modèle de *Blinn-Phong* à la surface *Lambertienne*. Dans ce cas, les trois variantes sont possibles, tel qu'illustré dans les figures 5.4b à 5.4d. Elles ont les mêmes avantages et inconvénients que pour le rendu de couleurs calculées.

3 Rendu de radiance

La radiance est définie sur un nombre fini de points de la surface (attaché aux sommets dans un premier temps). Nous souhaitons définir une interpolation cohérente afin de l'étendre à tout point de la surface. Celle-ci doit pouvoir correctement interpoler la couleur ainsi que les effets directionnels inclus dans la radiance.

3.1 Interpolation classique

Nous allons pour l'instant omettre les problèmes d'interpolation liés aux repères locaux : on suppose que les fonctions de radiance sont définies dans un repère unique et que l'interpolation de fonction revient donc, du fait de la linéarité de la base,

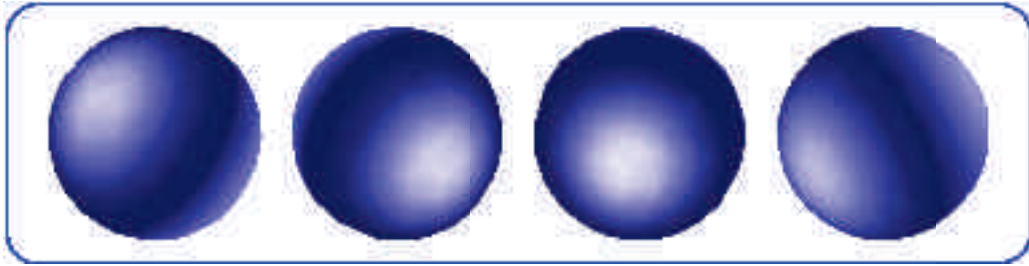


Figure 5.5 : *Défauts de l'interpolation scalaire, dite « classique ».*

à interpoler les coefficients. Nous traiterons le problème des repères locaux dans la section 3.3. Afin de faciliter les explications, les illustrations de cette section sont en deux dimensions.

Tous les travaux mettant en œuvre des champs de lumière se doivent de les visualiser. Dans le cas du champ de lumière plénoptique, un ensemble de plus proches « rayons » (*i.e.* des couleurs) sont directement interpolés dans un espace à 4 dimensions (deux spatiales et deux directionnelles : voir chapitre 3) [Gortler *et al.*, 1996; Levoy et Hanrahan, 1996; Davis *et al.*, 2012]. Il en va de même pour les champs de lumière surfaciques définis globalement [Miller *et al.*, 1998].

Dans les travaux traitant des champs de lumière surfaciques définis localement comme dans nos travaux, la radiance est définie par une fonction directionnelle placée sur la surface. Le rendu habituellement opéré est également une interpolation de scalaires : les fonctions en chaque sommet sont évaluées en fonction du point de vue courant et les scalaires résultants sont interpolés bi-linéairement sur la surface [Wood *et al.*, 2000; Coombe *et al.*, 2005; Jachnik *et al.*, 2012]. En termes d'interpolation fonctionnelle, ceci revient à définir la radiance d'un fragment p par interpolation (bi-)linéaire des fonctions définies aux sommets incidents à la face d'appartenance de p :

$$R_p(\omega_r) = \alpha R_1(\omega_r) + (1 - \alpha) R_2(\omega_r) \quad (5.1)$$

où R_1 et R_2 représentent les fonctions de radiance des sommets (ici en 2D).

L'avantage de ce mode de rendu est sa simplicité. Évaluer la radiance en p par l'équation (5.1) est équivalent⁷ à évaluer la fonction en chaque sommet puis à interpoler la couleur résultante. En ce sens, l'évaluation est simple à implémenter car l'interpolation des scalaires peut être opérée par le matériel graphique.

L'inconvénient de ce mode de rendu est son manque d'illusion de continuité d'ordre supérieur. On peut voir une analogie entre ce rendu et le lissage de Gouraud étudiée dans la section 2.2 : il s'agit dans les deux cas d'une interpolation de couleurs par sommet. Ce rendu exhibe ainsi les mêmes défauts. La figure 5.5 est le résultat d'un rendu de maillage avec radiance synthétique (de type diffus + spéculaire) aux sommets et rendu par cet algorithme. On peut y deviner le maillage triangulaire sous-jacent et les reflets spéculaires sont mal rendus (tâchés de fragments plus

⁷. À condition que les vecteurs *point de vue-sommet* soient proches pour les trois sommets du triangle.

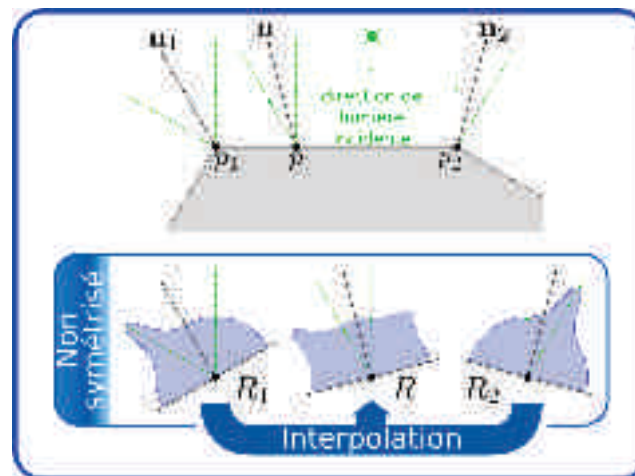


Figure 5.6 : Radiance résultante après considération des hypothèses (1) et (2). La radiance R_1 en p_1 exhibe un pic dirigé selon l'axe symétrique à la lumière incidente ω_i par rapport à la normale \mathbf{n}_1 (i.e. vers la gauche) alors que la radiance R_2 en p_2 exhibe un pic dirigé selon l'axe symétrique par rapport à \mathbf{n}_2 (i.e. vers la droite). Interpoler ces deux fonctions en atténue les reflats spéculaires et ne correspond aucunement à la radiance telle qu'elle aurait été définie en un point intermédiaire p .

sombres).

Ce rendu génère souvent de mauvais résultats. Pour expliquer cela, nous supposons que les deux hypothèses suivantes sont vérifiées :

Hypothèse 1 : La réflectance exhibe un comportement de type couleur diffuse et réflexion spéculaire orientée dans le sens de réflexion idéal (i.e. l'axe symétrique autour de la normale par rapport au rayon de lumière incident).

Ceci est vrai pour de nombreux matériaux dans le monde réel : mobilier, sol, la plupart des peintures, etc. Il existe bien évidemment des matériaux ne la respectant pas : peintures spéciales (e.g. bandes d'autoroute), certains métaux, etc.

Hypothèse 2 : L'environnement lumineux présent lors de l'acquisition est suffisamment éloigné pour être assimilé à des sources ponctuelles placées à l'infini.

Pour deux sommets voisins, cette hypothèse est généralement vraie. Des cas particuliers (occultation de source lumineuse) peuvent ponctuellement la mettre en défaut.

Le résultat de la radiance en un point du maillage est alors tel qu'illustré dans la figure 5.6 : la lumière est réfléchi de façon uniforme dans toutes les directions, auquel est ajouté un pic spéculaire orienté vers la gauche (pour le sommet de gauche) ou à droite (pour le sommet de droite). Plus les normales de ces sommets sont écartées, plus les pics spéculaires s'écartent (du double de l'angle entre les normales).

La radiance interpolée en un point intermédiaire d'après l'équation (5.1) est composée de la même couleur diffuse omnidirectionnelle, agrémenté de deux pics atténués (voir courbe de R dans la figure 5.6). Or, la radiance aurait été toute autre si elle avait été mesurée en ce point (voir courbe solide). Ceci n'est donc pas une interpolation correcte.

3.2 Interpolation de radiance « symétrisée »

Notre objectif est de définir l'interpolation déterminant la radiance R en un point intermédiaire p de façon à interpoler les fonctions de radiance R_1 et R_2 sans perte ou atténuation des reflets. Elle doit idéalement être identique à la radiance simulée en p sous les hypothèses (1) et (2).

Un traitement se basant sur les deux hypothèses sus-mentionnées permet de rendre les radiances comparables : la symétrisation [Wood *et al.*, 2000].

Définition

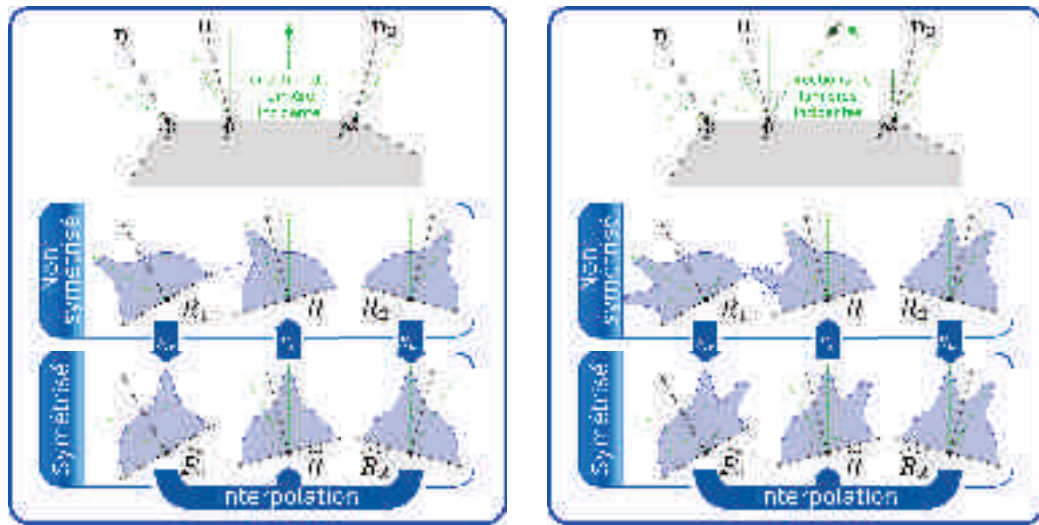
Symétrisation . Soit $R(\omega)$ une fonction de radiance définie sur l'hémisphère orienté par sa normale n . La fonction symétrisée $\widetilde{R}(\omega_r) = R(S_{\mathbf{n}}\omega_r)$ est la symétrique de $R(\omega_r)$ par rapport à \mathbf{n} : il s'agit d'une rotation d'angle π radians autour de l'axe \mathbf{n} .

La figure 5.7 montre que les versions symétrisées des radiances ont des pics spéculaires orientés dans la même direction (celle de la source). En raison de l'additivité des contributions lumineuses (voir chapitre 2, page 35), ceci se généralise à un environnement lumineux quelconque, à condition de respecter les hypothèses (1) et (2), résultant ainsi en une fonction de radiance avec de multiples pics spéculaires.

L'interpolation peut alors se faire comme suit (voir figure 5.7). Les radiances R_1 et R_2 des sommets p_1 et p_2 sont symétrisées autour de leurs normales respectives \mathbf{n}_1 et \mathbf{n}_2 : $\widetilde{R}_1(\omega_r) = R_1(S_{\mathbf{n}_1}\omega_r)$ et $\widetilde{R}_2 = R_2(S_{\mathbf{n}_2}\omega_r)$. Ensuite, ces fonctions \widetilde{R}_1 et \widetilde{R}_2 sont interpolées au point d'intérêt $p = \alpha p_1 + (1 - \alpha)p_2$, tout comme la normale $\mathbf{n} = \alpha \mathbf{n}_1 + (1 - \alpha)\mathbf{n}_2$. La fonction interpolée $\widetilde{R} = \alpha \widetilde{R}_1 + (1 - \alpha)\widetilde{R}_2$ est alors à nouveau symétrisée autour de la normale interpolée : $R(\omega_r) = \widetilde{R}(S_{\mathbf{n}}\omega_r)$. Au final, l'interpolation de fonctions de radiance est formellement définie par :

$$R(\omega) = \alpha R_1(S_{\mathbf{n}_1}S_{\mathbf{n}}\omega_r) + (1 - \alpha)R_2(S_{\mathbf{n}_2}S_{\mathbf{n}}\omega_r) \quad (5.2)$$

Nous verrons dans les résultats que cette définition est bénéfique pour nos cas synthétiques et réels : le gain en réalisme est analogue au passage du lissage de *Gouraud* à l'interpolation de *Phong* sur les couleurs.



(a) Cas simple : 1 pic induit par une source lumineuse.

(b) Généralisation : 2 pics induits par deux sources lumineuses.

Figure 5.7 : *Symétrisation de la radiance : sous les hypothèses (1) et (2), les fonctions symétrisées \bar{R}_1 et \bar{R}_2 sont similaires même lorsqu'elles sont orientées différemment. Les interpoler puis re-symétriser permet de correctement définir une fonction de radiance \tilde{R} en tout point p : les reflets spéculaires sont correctement orientés.*

3.3 Implémentation

Évaluer l'équation (5.2) requiert de pouvoir calculer un terme $\bar{R}_i(\omega) = R_i(S_{\mathbf{n}_i} S_{\mathbf{n}_\omega})$ pour toute fonction à interpoler afin d'interpoler les coefficients par la suite. Ceci implique d'appliquer *deux symétrisations* sur chaque fonction et de savoir *interpoler analytiquement*.

Dans le cas d'une définition en repère global, les symétrisations sont calculables analytiquement : elles sont équivalentes à une rotation d'angle π autour de la normale. Ce problème a été étudié notamment pour les harmoniques sphériques : les formules sont connues même si fastidieuses à implémenter [Gimbutas et Greengard, 2009]. Par contre, évaluer ces rotations sur GPU est d'une complexité rédhibitoire : le calcul n'est pas réalisable en temps réel.

Inversement, si R est définie dans un repère local, une solution analytique n'est pas garantie, en particulier dans le repère cartésien projeté de la base polynomiale. La rotation autour de la normale de la fonction correspond à un simple changement de variables (inversion des vecteurs de la base orthogonale) $S(u, v) = (-u, -v)$. La rotation autour d'autres axes est mal définie dans un tel repère : il est impossible de représenter la même fonction dans un repère 2D paramétré sur un autre plan tangent.

3.3.1 Solutions techniques

Grâce à l'égalité $\widetilde{R}_i(\omega) = R_i(S_{n_i}\omega)$, nous contournons ce problème en ré-écrivant la formule d'interpolation (5.2) :

$$R(\omega) = \alpha\widetilde{R}_0(S_{\mathbf{n}}\omega_r) + \beta\widetilde{R}_1(S_{\mathbf{n}}\omega_r) + (1 - \alpha - \beta)\widetilde{R}_2(S_{\mathbf{n}}\omega_r) \quad (5.3)$$

Évaluer cette formule permet d'éviter l'interpolation analytique des fonctions et les rotations complexes sur GPU. Désormais, pour évaluer l'équation 5.2 dans le *fragment shader*, il suffit en effet de symétriser les directions d'évaluation et d'évaluer l'équation (5.3) pour un fragment p quelconque. Pour cela, le *fragment shader* a besoin des éléments suivants en entrée :

- ω_r , la direction d'évaluation,
- \mathbf{n} , la normale interpolée en p ,
- $\widetilde{R}_{\{0,1,2\}}(\omega)$, les trois fonctions symétrisées,
- $(\alpha, \beta, 1 - \alpha - \beta)$, les coordonnées barycentriques du fragment dans le triangle.

Pour ce faire, la fonction et la normale (voire le repère de *Frenet* dans le cas local) sont passés à la carte graphique en tant qu'attributs de sommet. Le vecteur ω_r se définit dans le *fragment shader* à l'aide du vecteur *point de vue*- p .

La normale interpolée s'obtient par normalisation d'une interpolation tri-linéaire. La fonction symétrisée \widetilde{R}_i est obtenue par un pré-calcul : la rotation se fait dans le logiciel et les paramètres sont passés en tant qu'attributs à la carte graphique.

Nous expliquons ci-dessous comment obtenir les trois fonctions et les coordonnées barycentriques en entrée du *fragment shader*. Pour cela, nous exploitons le *geometry shader* qui permet d'avoir accès aux attributs liés aux trois sommets composant un triangle et de les modifier. Nous faisons en sorte que l'interpolation faite par la carte graphique génère les attributs souhaités après la rasterisation.

Le pseudo-code du *geometry shader* est donné dans l'algorithme 4 où des éléments triviaux ont été omis : données liées au plongement géométrique, au traitement des trois canaux et à la gestion du repère local. Un triplet d'attributs de position, de normale et de radiance symétrisée (pré-calculée) sont fournis en entrée. En sortie, on souhaite émettre trois sommets, chacun munis d'une position, d'une normale et d'une coordonnée barycentrique interpolée ainsi que *des trois* fonctions de radiance symétrisées non interpolées. Le corps du *geometry shader* fait alors les manipulations nécessaires afin d'émettre ces trois fonctions pour les trois sommets (lignes 14 à 18). Ensuite, chaque sommet se voit associer sa position et sa normale d'origine ainsi que la coordonnée barycentrique qui lui correspond (lignes 19 à 23, 24 à 28 et 29 à 33 respectivement).

Ceci produit l'effet suivant après la rasterisation. Tout fragment reçoit en entrée trois fonctions de radiance symétrisées non interpolées, une position, une normale et des coordonnées barycentriques interpolées. Le *fragment shader* n'a alors plus

Algorithme 4 *Geometry shader* préparant les données.

```
01 // En entrée: primitive triangle
02 in vec3 p[]; // positions
03 in vec3 n[]; // normales
04 in s-RF R[]; // fonctions de radiance symétrisées

05 // En sortie: primitive triangle
06 // À tout fragment, on envoie :
07 smooth vec3 position; // une position interpolée
08 smooth vec3 normal; // une normale interpolée
09 smooth vec3 baryCoords; // des coordonnées barycentriques
10 flat out s-RF R0, R1, R2; // 3 fonctions de radiance
11 // symétrisées non interpolées

12 void main()
13 {
14 // les 3 fonctions de radiance
15 // à transmettre à tout fragment
16 R0 = R[0];
17 R1 = R[1];
18 R2 = R[2];

19 // sommet 1
20 position = p[0];
21 normal = n[0];
22 baryCoords = vec3(1,0,0);
23 EmitVertex(); // envoi du sommet 1

24 // sommet 2
25 position = p[1];
26 normal = n[1];
27 baryCoords = vec3(0,1,0);
28 EmitVertex(); // envoi du sommet 2

29 // sommet 3
30 position = p[2];
31 normal = n[2];
32 baryCoords = vec3(0,0,1);
33 EmitVertex(); // envoi du sommet 3

34 EndPrimitive(); // Fin du triangle
35 }
```

qu'à déterminer la symétrisation du vecteur ω_r autour de la normale interpolée et d'évaluer l'équation (5.3) pour déterminer la couleur finale (voir algorithme 5).

Algorithme 5 *Fragment shader* pour l'évaluation de la formule (5.3).

```
// En entrée:
Position p; // interpolée
Normale n; // interpolée
CoordBarycentriques barycoords;
Fonctions R0, R1, R2;

void main()
{
    vec3 Wr = posCam - p;
    vec3 Snw = 2*(w.n)*n - w; // symétrisation autour de n

    FRAG_COLOR = barycoords.x * R0(Snw);
    FRAG_COLOR += barycoords.y * R1(Snw);
    FRAG_COLOR += barycoords.z * R2(Snw);
}
```

Ce mécanisme évite ainsi d'avoir à interpoler des fonctions à proprement parler. Le coût lié à cette opération sera discuté dans la section 3.3.2.

3.3.2 Détails et commentaires

Vocabulaire

Interpolation scalaire : interpolation des scalaires après évaluation des fonctions (équivalent à la formule (5.1)).

Interpolation fonctionnelle : interpolation des fonctions avant leur évaluation (équivalent à la formule (5.2) et mis en œuvre par la formule (5.3)).

Débordement du domaine. La surface représentée par un maillage est linéaire par morceaux. Ainsi, une fonction de radiance attachée à un sommet peut être visualisée en-dehors de son hémisphère de définition : dans le cas où la surface est localement convexe. Dans ce cas et si la fonction est définie sur un domaine hémisphérique, alors la couleur peut être indéfinie pour certains points de vue. Pour résoudre cela, nous prolongeons la valeur au bord du domaine le long des méridiens afin d'assurer une visualisation continue lors du passage du bord du domaine.

Performances. L'évaluation de trois fonctions par *shader* plutôt qu'une seule comporte un sur-coût (un facteur 3 au moins). Cependant, rapporter des temps de calcul d'une image ne serait pas très utile : dans le cas de la formule (5.1) (évaluation par sommet) la complexité est liée au nombre de sommets du maillage, alors que dans celui de la formule (5.2) (évaluation par fragment), elle est liée au nombre de pixels affichés à l'écran et couvrant l'objet.

Une réflexion intéressante est la suivante : lorsque l'objet est visualisé de loin, un pixel couvre souvent plusieurs triangles, et ainsi, il y a moins d'évaluations à faire lorsque l'évaluation se fait dans le *fragment shader* que lorsque toute l'évaluation se fait une fois par sommet. Inversement, lorsque l'objet est visualisé de près, l'écran est saturé de pixels couvrant l'objet, et les temps de calcul sont alors augmentés. Or, dans ce cas précis, ce sur-coût est bénéfique en réalisme car il s'agit de la configuration dans laquelle un triangle couvrira au moins une dizaine de pixels : évaluer plus finement que par sommet permet de gagner en réalisme.

Optimisations. Le passage des fonctions de radiance (*i.e.* leurs coefficients et leur repère local) à travers les étapes du GPU est impossible lorsque la taille de la base est grande : le nombre d'attributs par sommet est limité par le matériel et la duplication des fonctions dans le *geometry shader* est lourde. Une astuce consiste alors à stocker ces données dans des tableaux (les textures GPU) et d'associer les indices aux sommets. Le nombre de textures GPU est également limité, mais cette astuce permet de passer d'une dizaine de coefficients par fonction à une centaine. Au-delà, il faudra avoir recours à des techniques dites de « *streaming* », chargeant et déchargeant la mémoire GPU en fonction des besoins. Le rendu temps réel devient alors impossible.

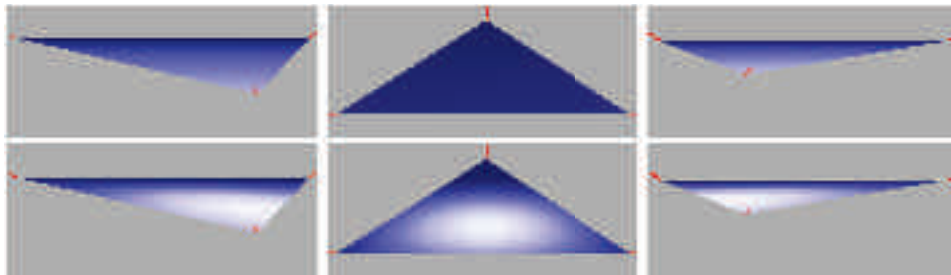
3.4 Résultats

L'interpolation définie par l'équation (5.2) génère le résultat souhaité : une interpolation préservant correctement la couleur diffuse et l'intensité accrue d'un pic spéculaire en tout fragment.

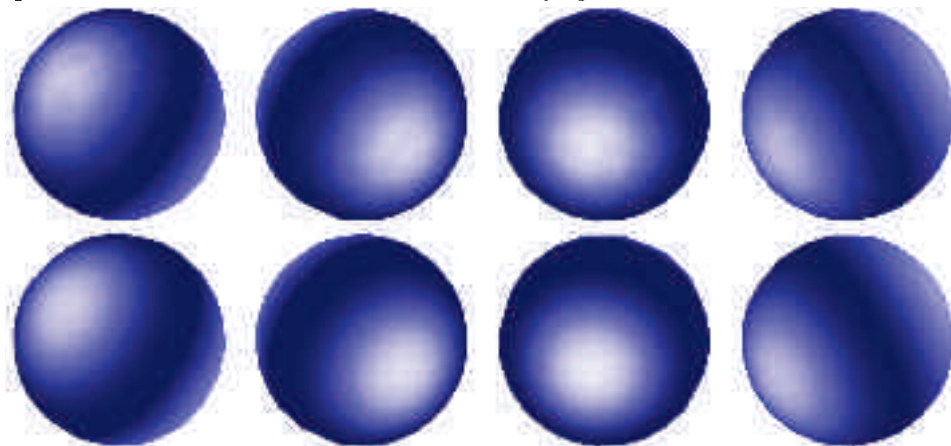
Jeu de test. La figure 5.8 illustre le gain par rapport à un rendu avec évaluation par sommet puis interpolation des couleurs. Il s'agit ici de données synthétiques qui sont le résultat d'une reconstruction à partir d'une acquisition simulée (à l'instar de l'objet *bunny* dans le chapitre 4). Le modèle de réflectance simulé est composé d'une couleur diffuse bleue et d'un effet spéculaire blanc.

L'objet *triangle* (figure 5.8a) est composé d'un seul triangle topologique. Une source lumineuse blanche est placée au-dessus du plan défini par le triangle. Les hémisphères des trois fonctions de radiance reconstruites sont orientées selon les normales illustrées en rouge dans la figure : les orientations sont très divergentes. La normale d'un

sommet forme un angle de 45° avec le plan du triangle et est orientée vers l'extérieur du triangle : *i.e.* elle est incluse dans le plan perpendiculaire au triangle et le coupant par la droite reliant son barycentre au sommet. Cette figure illustre un cas regroupant des cas de figure proches de ce qui est illustré dans les figures 5.6 et 5.7.



(a) Rendus d'un seul triangle avec radiances stockées au sommet sans (haut) et avec (bas) interpolation des fonctions symétrisées. Les normales (fortement écartées) sont illustrées en rouge. Sans interpolation, de légers reflets spéculaires sont visibles sur un des trois sommets, selon les points de vue. Avec interpolation, le reflet spéculaire est correctement reconstruit en tout fragment.



(b) Rendus d'une sphère avec radiances stockées au sommet sans (haut) et avec (bas) interpolation des fonctions. Sans interpolation des fonctions, la structure triangulaire est visible. Avec interpolation, on retrouve l'illusion de la surface lisse.

Figure 5.8 : Comparaisons entre rendus sans (lignes du haut) et avec (lignes du bas) interpolation des fonctions avant leur évaluation.

Pour l'objet *sphère* (figure 5.8b), deux sources lumineuses ponctuelles blanches ont été placées au-dessus de ses deux pôles. Les normales d'un sommet y sont le résultat d'une interpolation des normales des faces adjacentes.

Discussion des résultats. La figure 5.8 montre tous les bienfaits de l'interpolation de fonctions de radiance. L'évaluation des fonctions de radiance aux sommets avant interpolation génère une atténuation des reflets aux fragments interpolés (*i.e.* éloignés d'un sommet). C'est le phénomène d'atténuation de fonctions de radiance mélangées (cf. figure 5.6) qui en est responsable.

La figure 5.8a illustre cette atténuation. Lorsque l'observateur est placé dans l'axe de réflexion spéculaire d'un des sommets, un pic spéculaire est visible (figures du haut, aux extrêmes gauche et droite) proche d'un sommet. Pour une observation à partir d'une position intermédiaire (centre), aucun reflet n'est visible. L'interpolation scalaire n'est en effet pas capable de reproduire les reflets correctement.

La ligne du bas de la figure 5.8a montre que l'interpolation de fonctions symétrisées permet de bien représenter les reflets en tout fragment. L'intensité accrue du pic spéculaire est préservée en tout fragment, même lorsque l'axe de visualisation n'est pas l'un des axes de réflexion idéaux d'un des trois sommets. Grâce à l'interpolation fonctionnelle, les trois fonctions sont évaluées en leur pic spéculaire lorsque cela est judicieux et pour tout fragment.

Un effet de bord de l'atténuation des pics lorsque la formule (5.1) est appliquée est la reconnaissance de la structure triangulaire d'un objet non densément maillé, visible dans la partie haute de la figure 5.8b. Les figures du bas illustrent que l'interpolation de fonctions symétrisées avant évaluation (formule (5.2)) permet d'éviter cet écueil. Avec une tessellation toute aussi peu dense, l'interpolation de fonctions symétrisées permet de donner l'illusion d'une sphère lisse et d'une réflexion de lumière continue.

Sur les données réelles, le gain est plus difficile à illustrer : les maillages reconstruits après numérisation sont tellement fins que l'interpolation s'avère peu utile (*i.e.* il

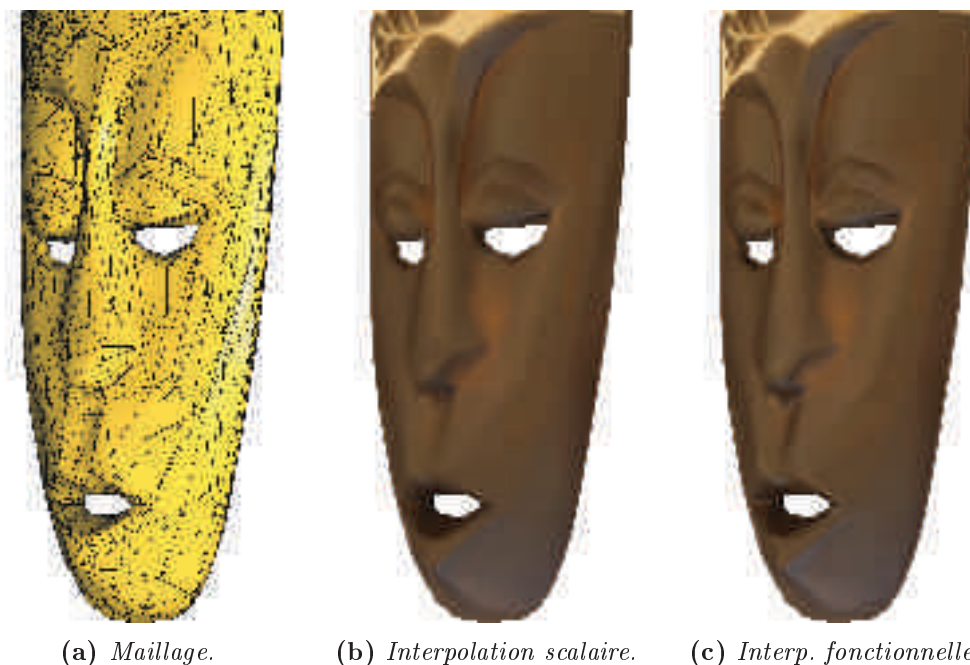


Figure 5.9 : Comparaisons entre rendus selon les formules (5.1) et (5.2) respectivement. Un certain nombre de reflets sont mieux rendus dans la figure 5.9c que dans la figure 5.9b. Ceci est particulièrement visible sur le menton, mais également sur les sourcils et le nez.

y a plus de sommets que de fragments à afficher). L'interpolation sera cependant capitale dans le chapitre suivant pour la visualisation de données simplifiées où de grands triangles peuvent apparaître. La figure 5.9 montre le maillage simplifié de la figure 6.9 (page 160) rendu avec et sans interpolation des fonctions de radiance. Ici aussi, on peut voir que les effets directionnels de la radiance sont mieux reconstruits dans les fragments éloignés des sommets : le reflet du menton est mieux rendu, celui du nez et de la paupière gauche n'apparaissent que dans la figure 5.9c et les reflets des sourcils sont également plus réalistes.

4 Conclusion

La nouvelle visualisation mise en place permet de passer d'un rendu de fonctions de radiance opéré de façon analogue au lissage de *Gouraud* à une réelle interpolation de *Phong* étendue aux fonctions. Pour cela, nous nous plaçons dans un cadre où les hypothèses (1) et (2) sont vérifiées (ce qui est souvent le cas) et avons recours aux fonctions symétrisées et à une ré-écriture de l'équation d'interpolation. Nous exploitons la flexibilité du *geometry shader* pour contourner les lourds calculs analytiques et implémenter le tout sur GPU.

Cette interpolation augmente le réalisme de façon considérable lorsqu'on dispose d'un maillage peu dense où des dizaines de pixels couvrent un triangle, ou si l'application a souvent recours à de forts zooms. Un corollaire est que le rendu permet d'obtenir un réalisme équivalent avec un maillage moins dense.

Les limites de cette méthode résident dans le coût de cette interpolation et dans les hypothèses.

Le sur-coût est lié au réalisme obtenu : un maillage moins dense pourra être rendu avec la même qualité mais à un coût calculatoire plus fort. Inversement, lorsque le maillage est dense, le nombre de sommets avoisine le nombre de pixels, et le sur-coût d'un rendu interpolé est faible. Ce sur-coût peut cependant atteindre un seuil rédhibitoire, dont la valeur dépend du matériel graphique, de la complexité de la scène et du taux de couverture des pixels-écran.

Les hypothèses que nous avons considérées sont souvent vérifiées. Lorsqu'elles ne le sont pas, les contributions de ce chapitre sont vaines, mais le résultat ne sera pas pire que si l'interpolation scalaire est opérée.

4.1 Bilan

Comme dans le chapitre précédent, nous avons pu fournir une solution utilisable dans de nombreux cas de figure. Ainsi, ce mécanisme peut être mis en place pour des fonctions de radiance exprimées dans un repère local ou global et quelque soit

la base utilisée et sa taille. On laisse ainsi toujours la totale liberté dans le choix du modèle à l'utilisateur.

On retiendra aussi que la symétrisation permet d'exploiter la cohérence spatiale : des radiances symétrisées voisines sont proches lorsque les hypothèses (1) et (2) sont vérifiés. Ceci sera d'une importance capitale lorsque l'on cherchera à comparer deux fonctions de radiance dans l'outil de simplification proposé dans le chapitre 6.

4.2 Perspective : radiance par texel

Nous souhaitons évoquer les travaux à entreprendre en vue d'un passage à une radiance stockée dans une texture.

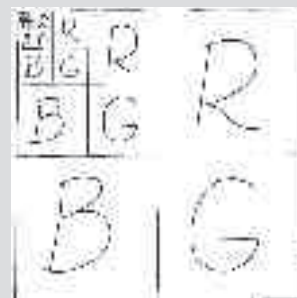
Lorsque l'information colorimétrique est stockée dans une texture, la résolution spatiale est également finie. Lors du rendu, il s'agit de calculer la couleur d'un pixel à l'écran, qui a une certaine empreinte sur la texture. Pour les explications qui suivent, nous distinguons deux cas : le cas où cette empreinte est de taille inférieure à un texel et l'inverse. Dans le premier cas, on souhaitera interpoler quelques texels (*e.g.* les quatre texels dont le point central est le plus proche de l'empreinte). Dans le second cas, il faut éviter l'*aliasing* (*i.e.* l'effet en « marches d'escalier ») et on souhaitera que la couleur du pixel résulte d'un mélange des texels inclus dans son empreinte. On parle alors de filtrage.

Filtrage. Il y a deux façons populaires de gérer l'*aliasing* : le mip-mapping et le filtrage « manuel ». Dans les deux cas, il s'agit de déterminer une moyenne des informations des texels inclus dans l'empreinte d'un pixel écran. Cette empreinte peut couvrir une portion de surface quelconque et il est ainsi difficile d'estimer ces moyennes, et le calcul analytique est – comme nous l'avons vu –, difficile du fait des repères locaux : il ne suffit pas de moyennner des coefficients de fonctions [Tan *et al.*, 2008; Dupuy *et al.*, 2013].

Dans le cas du *mip-mapping*, il s'agit de pré-calculer ces moyennes.

Définition

Mip-map [Williams, 1983] : structure définissant une texture à différentes résolutions réduites successivement de moitié. Les textures grossières sont généralement calculées en moyennant des texels quatre par quatre de la texture plus fine d'un niveau.



Il s'agit de définir des textures de différentes résolutions, à savoir la texture originale puis un certain nombre de textures de tailles successivement divisées

par deux. Ainsi, lors du rendu, on choisira la texture dont la résolution est la plus adaptée pour se retrouver dans le cas de figure où l’empreinte du pixel écran est de taille inférieure à celle d’un texel. On se ramène ainsi à un cas d’interpolation. En réalité, on considère les deux résolutions de texture telles que l’empreinte couvre partiellement 4 texels d’un niveau et 1 texel du niveau plus grossier et l’interpolation sera ainsi tri-linéaire.

La difficulté est de pré-calculer une *mip-map* pertinente. Pour de simples valeurs de couleurs, cela peut se faire en moyennant les couleurs quatre par quatre. Pour les fonctions de radiance, il s’agit de définir des fonctions de radiance interpolées (*i.e.* des coefficients) qui approchent quatre fonctions d’un niveau plus fin. Une interpolation scalaire ne donne pas le résultat souhaité, comme nous l’avons vu dans ce chapitre.

Il serait cependant envisageable de moyenner en exploitant l’interpolation que nous avons définie dans ce chapitre (équation (5.2) étendue à l’interpolation de quatre valeurs de texels au lieu de trois valeurs de sommets). Ce calcul est faisable analytiquement dans le cas d’un repère global (il suffit d’interpoler les coefficients de fonctions symétrisées). Dans le cas d’un repère local cependant, nous avons vu qu’une interpolation analytique est complexe voire impossible : cela dépend de la base de fonctions.

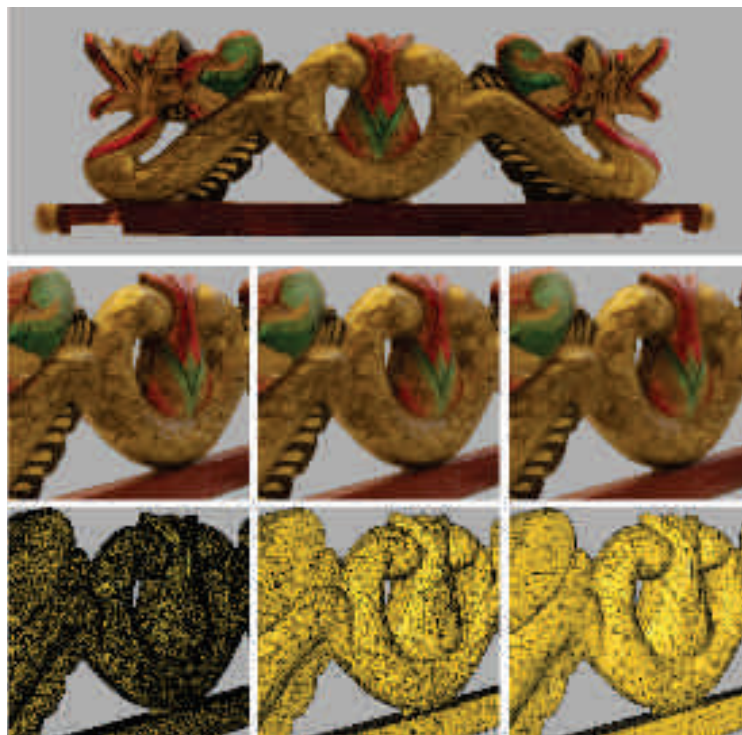
Dans le cas du filtrage dit « manuel », le calcul de l’interpolation des informations des texels inclus dans l’empreinte d’un pixel se fait explicitement lors du rendu (*i.e.* dans le *pixel shader*). Il faut pour cela désactiver le filtrage matériel et faire les accès textures adéquats (*i.e.* récupérer tous les texels inclus dans l’empreinte d’un pixel écran) en vue d’une interpolation selon une formule pertinente implémentée dans le *fragment shader*. Même si une formule peut être définie en respectant la logique des fonctions de radiance symétrisées, les temps de calculs seront lourdement impactés : lorsque la taille de l’empreinte devient trop grande, il faut alors faire de nombreux accès textures pour chaque pixel, ce qui ralentit grandement le rendu.

Interpolation. L’interpolation est nécessaire afin d’obtenir un résultat lisse. L’empreinte de pixel tombant entre quatre texels (voire cinq dans le cas de *mip-maps*), il suffit de les interpoler. Ce mécanisme est défini en dur dans le matériel graphique et est donc efficace. Or, il ne fait qu’interpoler les coefficients de fonction. Contrairement au pré-calcul de *mip-map*, il ne s’agit ici que d’évaluer une fonction interpolée, et non de la définir pour la stocker. Ainsi, on peut appliquer l’équation (5.3) (étendue à plusieurs fonctions), qui permet de gérer le cas du repère local.

En somme, nos travaux d’interpolation sont théoriquement transposables pour l’interpolation de textures de radiance. La gestion de l’*aliasing* par le pré-calcul de *mip-map* n’est cependant faisable que dans le cas où les fonctions s’expriment toutes dans le même repère (global). La gestion de fonctions exprimées en repères locaux nécessite davantage d’attention avant d’obtenir une solution fonctionnelle.

Chapitre 6

Simplification de maillage surfacique avec attribut de radiance



RÉSUMÉ:

Nous abordons ici la simplification du champ de lumière surfacique tel que nous le représentons. Les maillages et les fonctions de radiance qui y sont attachés peuvent être volumineux et il y a ainsi un réel besoin de pouvoir simplifier les données. Pour ce faire, nous survolons les méthodes de simplification de maillage et analysons les critères de simplification portant sur des attributs scalaires et vectoriels. Nous y constatons également qu'aucune méthode ne traite d'attributs fonctionnels. Nous proposons ensuite un nouveau critère applicable dans le cadre de la simplification d'un maillage avec attribut de couleur par sommet via un algorithme de contractions d'arêtes. Ce critère a la particularité de considérer la façon dont le maillage est rendu en tout point (et non seulement sur les sommets) et d'être facilement extensible à des fonctions de radiance. Ceci nous permet finalement de proposer un critère portant sur les fonctions de radiance. Il estime la perte qualitative de la contraction d'une arête par une différence fonctionnelle entre la radiance du sommet qui est amené à disparaître et la radiance qui sera le résultat de l'interpolation en ce point après contraction. Nous montrons finalement des résultats qualitatifs démontrant le besoin d'une métrique sur les fonctions de radiance dans un premier temps et des résultats de simplifications satisfaisantes ensuite. Nous discutons également de l'extensibilité aux maillages progressifs et au raffinement adaptatif.

Sommaire

1	Simplification de maillages : une taxinomie	139
2	Algorithme de contraction d'arêtes	141
2.1	Critères géométriques	142
2.1.1	QEM : métrique d'erreur	142
2.1.2	QEM : plongement de sommet contracté	144
2.2	Critères couleur	145
2.2.1	Extension QEM dans \mathbb{R}^{3+m}	145
2.2.2	Extension QEM dans \mathbb{R}^3 et \mathbb{R}^m	146
2.3	Bilan de l'existant	147
3	Nouveau critère combinant géométrie et couleur	149
3.1	Critère sur la couleur	150
3.2	Calcul du gradient	152
3.3	Résultats qualitatifs	152
4	Critère combinant géométrie et radiance	154
4.1	Critère sur la radiance	155
4.2	Calcul de l'erreur	156
4.3	Résultats qualitatifs	156
4.3.1	Données synthétiques	157
4.3.2	Modèles numérisés	159
5	Conclusion	159
5.1	Bilan	159
5.2	Perspectives	162
5.3	Discussion : contraction avec plongement optimal	163

Nous avons pu nous apercevoir dans ce manuscrit que la visualisation d'un champ de lumière surfacique nécessite une grande quantité de calculs : des milliers voire des millions d'évaluations de fonctions sont nécessaires. La complexité mémoire est liée au nombre de sommets et de fonctions alors que la complexité calculatoire dépend du nombre de fonctions, de la complexité d'évaluation de la base choisie et du choix de la méthode d'interpolation (voir chapitre précédent).

Nous proposons ici un outil de simplification de données qui permet de créer un maillage avec radiance de taille réduit et qui approche au mieux le modèle original. Ceci permet de donner à l'utilisateur le choix du compromis entre qualité et complexité du modèle à manipuler.

Un tel outil est une solution adéquate dans de nombreux cas où la réduction du coût mémoire et calculatoire est bénéfique. Dans une scène composée de plusieurs objets, tous ne sont pas en avant-plan : un modèle moins complexe suffit alors pour représenter les objets qui seront petits car visualisés en arrière-plan. Il en va de même lorsque l'on visualise un modèle sur un écran de petite taille (*smart-phones*, tablettes, lecteurs vidéo, etc.). Dans ces cas, une faible dégradation sera quasiment imperceptible.

Dans d'autres cas, la capacité de calcul peut imposer de réduire la taille du modèle et la qualité visuelle peut en être impactée. Nous verrons que l'outil mis en œuvre peut s'étendre à des applications permettant du *streaming* adaptatif par le biais de maillages progressifs.

1 Simplification de maillages : une taxinomie

Il existe deux types de méthodes pour générer des maillages à différents niveaux de résolution [Heckbert et Garland, 1997].

La première procède par raffinement d'un modèle grossier jusqu'à obtention de la résolution souhaitée. Cette catégorie de méthodes a été très étudiée lors de l'essor de la *modélisation* graphique dans les années 1980.

La seconde méthode procède par simplification de maillages fins. Cette catégorie a été très étudiée dans les années 1990 suite à la commercialisation d'outils de vision et de *numérisation* 3D produisant des données volumineuses (*i.e.* scanners, stéréovision, etc.). Nous nous situons dans ce second cas. Nous allons catégoriser les outils de simplification de maillages afin d'aboutir au choix de l'outil le plus utilisé pour les maillages avec attributs : la simplification par contraction d'arêtes. Nous étudions ensuite cet algorithme en détail et l'adaptions à nos besoins.

Une catégorisation possible est la suivante :

La simplification volumique subdivise l'espace compris dans la boîte englobante d'un objet en sous-éléments (cubes, sphères ou autres). La densité d'échan-

tillonnage (*i.e.* la quantité d'éléments) est locale et dépend de la variation de la surface [Rossignac et Borrel, 1993; Low et Tan, 1997]. Un seul sommet est ensuite retenu pour chaque élément qui en contient en général plusieurs.

La simplification surfacique globale vise à créer une surface la plus proche possible de l'originale étant donnée une résolution fixée et réduite. Dans l'article de Turk [1992] par exemple, un nombre arbitraire de sommets est réparti sur la surface selon une méthode de relaxation itérative minimisant un critère global d'écart à la surface originale.

Simplification surfacique locale. Il s'agit d'outils simplifiant itérativement une ou quelques éléments topologiques. Parmi ces méthodes, les techniques de type *trouer et remailler* suppriment dans un premier temps un sommet [Schroeder *et al.*, 1992], un triangle [Hamann, 1994] ou un *patch* (*i.e.* quelques triangles) [Kalvin *et al.*, 1991] dûment sélectionnés. Les trous créés sont ensuite remaillés localement.

La technique de *contraction* d'arêtes permet d'éviter le remaillage : une arête est contractée en un point et ses éléments incidents s'y rejoignent désormais. La surface reste ainsi fermée (voir figure 6.1).

Nous choisissons d'utiliser la technique surfacique locale de contraction d'arêtes. Les avantages d'une technique locale sont les suivants.

- **Simplification progressive.** Procéder par contractions successives permet de définir précisément le compromis souhaité : chaque étape n'est qu'une modification infime du maillage.
- **Contrôle des attributs.** Il est pratique de pouvoir gérer le passage local de quelques attributs à un seul. Les attributs de ce dernier devront être représentatifs de ses prédécesseurs. Les techniques globales sont peu adaptées à la gestion d'attributs autre que la géométrie. Elles nécessiteraient en effet de définir en un seul calcul de nouvelles propriétés à associer à des sommets représentatifs d'un grand nombre de prédécesseurs.
- **Contrôle topologique.** La gestion locale permet de préserver la topologie, comme la caractéristique d'*Euler* par exemple¹.

Parmi elles, nous choisissons la seule contraction d'arête, en oppositions aux algorithmes composant plusieurs types de contractions. Ceci ne réduit pas la qualité du résultat et permet une transition douce : une contraction d'arête est peu visible et le *geomorphing* est possible [Hoppe, 1996].

Vocabulaire

Geomorphing : désigne la transition douce et progressive que l'on peut associer à une contraction d'arêtes en introduisant un rétrécissement progressif de l'arête jusqu'à convergence en un sommet.

1. Dans l'article de Garland et Heckbert [1997], la rigidité topologique est considérée comme un obstacle : il peut être plus utile de connecter des parties séparées afin de maintenir un meilleur aspect visuel de l'objet.

2 Algorithme de contraction d'arêtes

L'algorithme de simplification par contraction d'arêtes [Hoppe *et al.*, 1993; Guéziec, 1995] est basée sur l'opération unitaire illustrée par la figure 6.1 et se déroule comme décrit dans l'algorithme (6).

Algorithme 6 Algorithme de contraction d'arêtes.

- 1: Trier les arêtes selon un critère de priorité
 - 2: **tant que** file d'arêtes non vide **faire**
 - 3: - Contracter l'arête en tête
 - 4: - Mettre à jour la priorité des voisins du sommet contracté
 - 5: **fin tant que**
-

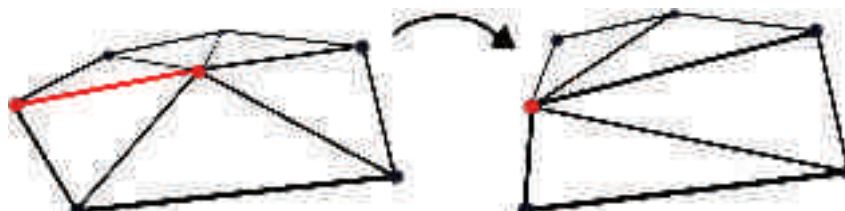


Figure 6.1 : *Opération unitaire de contraction d'arête. Deux sommets sont fusionnés en un seul, l'arête les reliant est dite « contractée » en un point, les deux faces incidentes disparaissent.*

Cet algorithme nécessite deux calculs afin d'associer à chaque arête :

1. l'erreur commise dans l'hypothèse de sa contraction ;
2. le plongement géométrique et les autres attributs à associer au sommet résultant de sa contraction.

À noter que le point 1 peut dépendre du point 2 : le placement et les attributs du nouveau sommet peuvent influencer sur l'erreur commise.

Des nombreux travaux ont portés sur la simplification de maillages par contraction d'arêtes. Seul le plongement géométrique est considéré dans un premier temps [Hoppe, 1996, 1997; Garland et Heckbert, 1997]. Ensuite ont été développés des critères prenant en compte des attributs scalaires ou vectoriels (couleur, normales, coordonnées de textures) [Garland et Heckbert, 1998; Hoppe, 1999] ou mesurant l'erreur en termes de couleur spécifiquement [Cohen *et al.*, 1998; Sander *et al.*, 2001; González *et al.*, 2007; Kim *et al.*, 2008]. Nous n'avons cependant pas identifié de travaux concernant la simplification de maillages avec attributs complexes, en particulier fonctionnels tel que des fonctions de radiance.

Nous allons toutefois analyser les critères de simplification connus. Les articles que nous explorons nous ont en effet donnés des aspects contextuels importants et les outils nécessaires à une appréhension correcte du problème de la simplification de maillages avec attribut de radiance.

2.1 Critères géométriques

Dans cette section nous considérons la simplification de maillages surfaciques plongés dans l'espace et sans autre attribut. Hoppe [1996] définit la priorité d'une arête en fonction de l'erreur géométrique d'écart à la surface commise dans l'hypothèse de sa contraction. Cette erreur est calculée grâce à une minimisation globale d'énergies mesurant la distance au maillage initial, le nombre de sommets et la longueur des arêtes. Elle a cependant été rapidement dépassée par la métrique d'erreur quadratique (QEM) [Garland et Heckbert, 1997] : une métrique efficace et fonctionnelle dont un des avantages est de pouvoir définir aisément un plongement dit « optimal » (*i.e.* qui minimise la métrique d'erreur). Afin de définir un critère sur les propriétés de radiance, nous utilisons la QEM en combinaison avec une métrique sur la radiance que nous définissons en section 4.

2.1.1 QEM : métrique d'erreur

Considérons la contraction d'arête $(v_0, v_1) \rightarrow \bar{v}$. L'erreur qui lui sera associée est une distance à la surface du maillage initial. Plus précisément, l'erreur commise pour un sommet contracté vaut la *somme des distances quadratiques aux plans définis par les triangles incidents à ses premiers ancêtres* (*i.e.* triangles du maillage non simplifié).

Somme des distances quadratiques aux plans incidents à un sommet.

Soient $\{\mathcal{P}_i\} : a_i x + b_i y + c_i z + d_i$ un ensemble de plans de normales unitaires $[a_i \ b_i \ c_i]$. Soient les triplets $\{A_{\mathcal{P}_i}, \mathbf{b}_{\mathcal{P}_i}, c_{\mathcal{P}_i}\} = \{[a_i \ b_i \ c_i]^T \cdot [a_i \ b_i \ c_i], d_i [a_i \ b_i \ c_i], d_i^2\}$ associés à ces plans.

Alors la somme des distances quadratiques d'un point p de coordonnées $[x \ y \ z]$ à ces plans s'écrit

$$d(p, \{\mathcal{P}_i\}) = [x \ y \ z] \left(\sum_i A_{\mathcal{P}_i} \right) \begin{bmatrix} x \\ y \\ z \end{bmatrix} + [x \ y \ z] \left(\sum_i \mathbf{b}_{\mathcal{P}_i} \right) + \left(\sum_i c_{\mathcal{P}_i} \right)$$

Métrique d'erreur quadratique. Soit $\{A_v, \mathbf{b}_v, c_v\} = \sum_{\mathcal{P} \in \hat{v}} \{A_{\mathcal{P}}, \mathbf{b}_{\mathcal{P}}, c_{\mathcal{P}}\}$ la somme des matrices des plans incidents au sommet v^2 . Alors l'erreur de la contraction $(v_0, v_1) \rightarrow \bar{v}$ vaut

$$\Delta((v_0, v_1) \rightarrow \bar{v}) = [\bar{v}_x \ \bar{v}_y \ \bar{v}_z] (A_{v_0} + A_{v_1}) \begin{bmatrix} \bar{v}_x \\ \bar{v}_y \\ \bar{v}_z \end{bmatrix} + [\bar{v}_x \ \bar{v}_y \ \bar{v}_z] (\mathbf{b}_{v_0} + \mathbf{b}_{v_1}) + (c_{v_0} + c_{v_1})$$

2. Le symbole \hat{v} désigne l'ombrelle du sommet v .

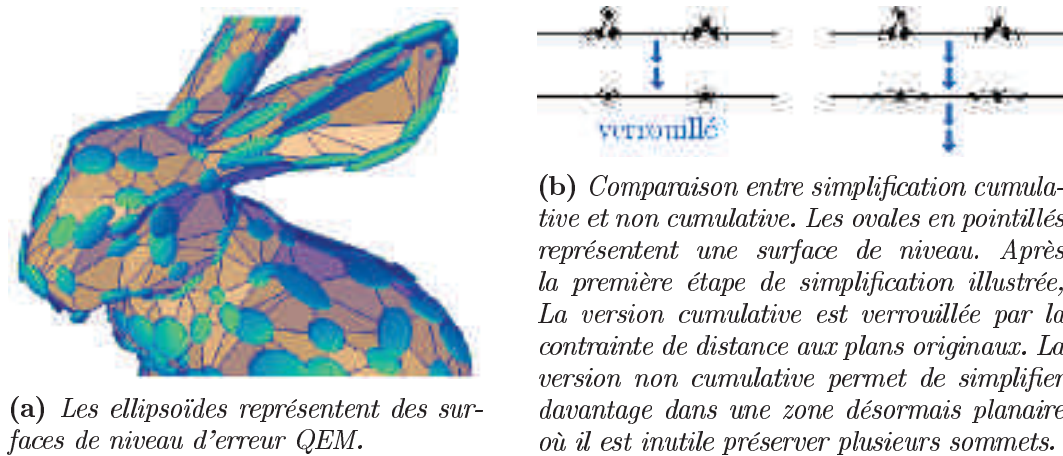


Figure 6.2 : Iso-contours d'erreurs QEM et variante non cumulative. Images de [Garland et Heckbert, 1998].

Remarquons qu'un plan peut être comptabilisé plusieurs fois dans ce calcul, les antécédents de deux sommets voisins ayant des triangles adjacents en commun. Cette erreur est négligée et impacte visiblement que peu les résultats.

Définition

Quadrique : $Q_v = \{A_v, \mathbf{b}_v, c_v\}$ est appelée la quadrique du sommet v .

Somme de quadriques : $Q_{v_1} + Q_{v_2} = \{A_{v_1} + A_{v_2}, \mathbf{b}_{v_1} + \mathbf{b}_{v_2}, c_{v_1} + c_{v_2}\}$.

Interprétation géométrique. La QEM possède une interprétation géométrique intéressante. En effet, les surfaces de niveau ou *iso-surfaces*³ forment généralement des ellipsoïdes. Ceci est illustré par la figure 6.2a. Le centre de l'ellipsoïde correspond au point minimisant l'écart à un ensemble de plans. Le long d'arêtes vives, ces ellipsoïdes sont plutôt allongées car le coût est moindre tant que l'on reste proche d'elles.

Cumul des erreurs. Afin de définir une erreur par rapport au maillage non simplifié, les auteurs proposent de cumuler les quadriques après contraction : $Q_{\bar{v}} = Q_{v_0} + Q_{v_1}$. Hoppe [1999] propose une variante, non cumulative⁴. Il propose de mesurer une distance par rapport au maillage courant (*i.e.* juste avant contraction). Les quadriques des sommets dont le voisinage a été modifié sont ainsi recalculées après chaque contraction au lieu d'être sommées.

Ceci peut paraître peu intuitif (on se plaît à croire qu'un écart au maillage original est

3. Ensemble des points de l'espace dont l'erreur $\Delta((v_0, v_1) \rightarrow \bar{v})$ est égale à une constante

4. Appelée « memoryless » dans son papier.

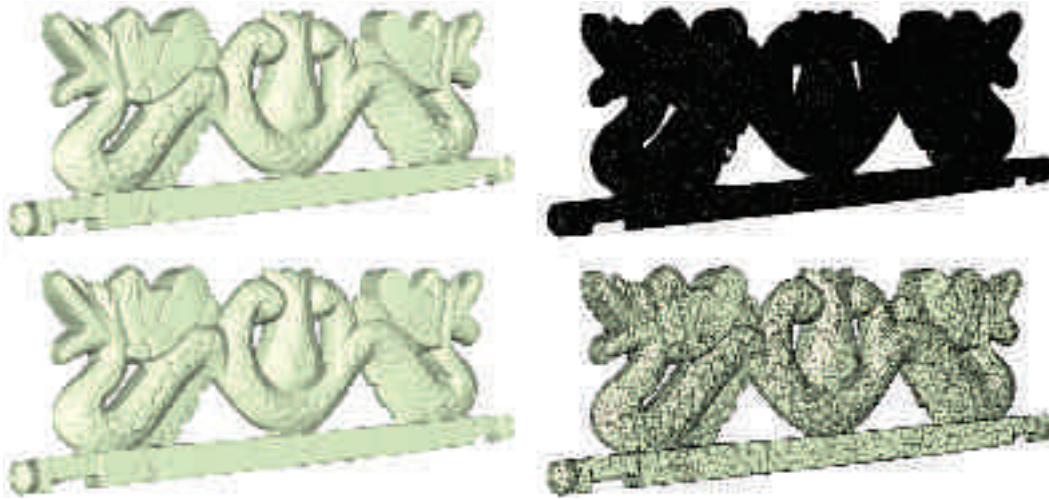


Figure 6.3 : *Simplification du double dragon avec la métrique QEM. Le modèle dense (en haut) est composé de presque 1 million de sommets alors que le modèle simplifié (en bas) de seulement 50 000.*

une meilleure métrique qu'un écart à un maillage simplifié), mais le résultat peut être meilleur, tel qu'illustré par la figure 6.2b. Dans la version cumulative (à gauche), les quadriques sont calculées une fois pour toute et sont sommées à la contraction. Ceci limite la liberté d'un sommet contracté tandis qu'une fois une propriété géométrique perdue (ici deux petites bosses), il est inutile de continuer à « préserver » certains sommets. La version non cumulative (à droite) recalcule les quadriques à chaque niveau et ne se soucie ainsi pas de « l'histoire » d'un sommet considéré.

2.1.2 QEM : plongement de sommet contracté

Finalement, il reste à définir quel plongement donner au nouveau sommet contracté \bar{v} . L'erreur $\Delta((v_0, v_1) \rightarrow \bar{v})$ est une expression quadratique en les coordonnées de \bar{v} . Ainsi, l'équation suivante annule le gradient et définit les coordonnées qui minimisent Δ :

$$\nabla(\Delta) = 0 \Leftrightarrow \frac{\partial \Delta}{\partial x} = \frac{\partial \Delta}{\partial y} = \frac{\partial \Delta}{\partial z} = 0$$

ce qui équivaut⁵ à résoudre :

$$\bar{v} = -\mathbf{b}_{\bar{v}}^T A_{\bar{v}}^{-1}$$

Cette formule n'est cependant applicable que si la matrice $A_{\bar{v}}$ est inversible (une non inversibilité sera caractérisée par des ellipsoïdes dégénérées). Dans le cas contraire, une solution de repli consiste à positionner \bar{v} à l'emplacement de v_1 , v_2 ou au milieu de l'arête ayant relié ces deux sommets.

5. Vérifiable en prenant les dérivées partielles sur la formule $\Delta((v_0, v_1) \rightarrow \bar{v})$.

La figure 6.3 illustre un exemple de simplification avec l'algorithme QEM. On peut voir que cet algorithme arrive à approcher des objets complexes (tels le *double dragon* que nous avons numérisé) avec un nombre faible de polygones.

2.2 Critères couleur

Suite à l'article de Garland et Heckbert [1997], plusieurs travaux ont présenté des métriques basées sur la QEM, étendant sa définition aux maillages avec attributs vectoriels tel que les normales ou la couleur. Parmi celles qui traitent de la couleur, nous présentons celles qui considèrent un attribut de couleur (souvent *RGB*) par sommet [Garland et Heckbert, 1998; Hoppe, 1999; Kim *et al.*, 2008] et définissent une métrique locale sur le maillage. Dans les perspectives, nous revenons sur celles qui traitent les couleurs par des textures et des mesures dans un espace de textures ou en pixels-écran [Cohen *et al.*, 1998; Sander *et al.*, 2001; González *et al.*, 2007].

2.2.1 Extension QEM dans \mathbb{R}^{3+m}

L'article de Garland et Heckbert [1998] est le premier à proposer une extension de la QEM aux attributs vectoriels. Les auteurs y évoquent en particulier les attributs couleur $(r, g, b) \in \mathbb{R}^3$ et coordonnées de texture $(u, v) \in \mathbb{R}^2$. Globalement, la portée de leurs travaux se limite aux attributs *linéairement interpolés* sur le maillage⁶.

Afin de préserver la possibilité de minimisation de l'erreur, les auteurs proposent de mélanger géométrie et autres attributs vectoriels ou scalaires dans des quadriques étendues, en se basant sur le principe de la QEM classique. Soit m la dimension des attributs autres que la géométrie. En considérant un plongement géométrique (dans \mathbb{R}^3) et un attribut de couleur (dans \mathbb{R}^3), on a donc $m = 3$.

Désormais, on se place dans l'espace $\mathbb{R}^{3+m} = \mathbb{R}^6$ afin d'appliquer les mêmes calculs qu'avec la QEM. Tout triplet de sommets définit alors un hyper-plan calculé à partir des coordonnées $[x \ y \ z \ r \ g \ b] \in \mathbb{R}^6$. L'erreur Δ est alors la même que pour la QEM classique, exploitant des quadriques étendues à la dimension 6. Le plongement optimal est également défini facilement.

Inconvénients. Premièrement, le *coût mémoire* grandit rapidement. Le nombre de réels à stocker au sein des quadriques est de complexité quadratique en m . Deuxièmement, la métrique choisie place dans un même espace des données de nature différente. Cet espace est donc non homogène. Alors que l'on souhaiterait pouvoir

6. Dans notre cas, l'interpolation est plus complexe que cela : voir chapitre 5.

mesurer une distance géométrique dans \mathbb{R}^3 et y ajouter des distances liés aux autres attributs, les mesures se font en \mathbb{R}^{3+m} . De plus, la non homogénéité des espaces revient à fixer une pondération entre les mesures sur chaque attribut. Par exemple, une mesure de distance dans un espace mélangeant de la géométrie (\mathbb{R}^3) et de la couleur dans l'espace RVB ($[0, 1]^3$) n'est pas invariante par mise à l'échelle : la géométrie sera prépondérante si les coordonnées géométriques sont grandes et inversement. En somme, la pondération est implicite, non contrôlée et donc *subie*.

2.2.2 Extension QEM dans \mathbb{R}^3 et \mathbb{R}^m

Dans [Hoppe, 1999; Hoppe et Marschner, 2000], il est proposé de mesurer une erreur similaire à [Garland et Heckbert, 1998], mais où l'espace géométrique est séparé de l'espace des attributs afin de mesurer une distance géométrique réelle dans \mathbb{R}^3 .

L'erreur totale à un sommet v est donnée par :

$$\Delta(v) = \Delta_g(v) + \sum_{j=1}^m \Delta_j(v) \quad (6.1)$$

où

- l'erreur géométrique $\Delta_g(v)$ est la distance euclidienne quadratique du sommet v à un ensemble de plans $\{\mathcal{P}_i\}$, tel la QEM initiale,
- l'erreur en attribut scalaire $\Delta_j(v)$ (avec $j \in [1, m]$) est l'écart quadratique entre l'attribut v_j et la valeur des attributs interpolés au point correspondant à la projection orthogonale de v sur chacun des plans \mathcal{P}_i . Elle est donc fonction du placement géométrique de v .

L'erreur géométrique quadratique peut s'écrire à l'aide de quadriques étendues à la dimension $m + 3$ ainsi :

$$\{A, b, c\}_g = \left\{ \left(\begin{array}{c|c} A_v & 0 \\ \hline 0 & 0 \end{array} \right), \left(\begin{array}{c} \mathbf{b}_v \\ 0 \end{array} \right), c_v \right\} \quad (6.2)$$

où les lignes séparatrices distinguent les 3 premières lignes et colonnes du reste de la matrice. L'erreur quadratique pour tout attribut j s'écrit également à l'aide de quadriques de même dimension⁷.

L'équation (6.1) peut s'écrire sous une forme quadratique en un vecteur de coefficients $[x \ y \ z \ a_1 \ \dots \ a_m]$ et à l'aide de trois matrices composant une quadrique. La minimisation est ainsi possible aussi facilement qu'auparavant : en résolvant un système linéaire. La pondération entre les éléments de nature différente est cependant toujours subie : elle est déterminée par l'ordre de grandeur des espaces respectifs des attributs.

⁷. Voir l'article de Hoppe [1999] pour connaître la façon dont ces matrices sont construites. *N.B.* : les lignes et colonnes 1 à 3 sont non creuses car Δ_j dépend du placement géométrique.

Autre métrique. Dans [Kim *et al.*, 2008], une autre métrique spécifique aux couleurs par sommet est proposée. Celle-ci n'est pas exprimée sous forme quadratique en ses coefficients, il n'est donc pas possible de la minimiser via la résolution d'une équation linéaire. Ainsi, seule la semi-contraction d'arête est considérée.

Définition

Semi-contraction : la contraction $(v_0, v_1) \rightarrow v_0$ ou v_1 . Le sommet résultant a la position et les attributs de l'un de ces deux ancêtres immédiats. Une semi-contraction est illustrée dans la figure 6.4.

Elle cumule l'erreur géométrique de la QEM avec une erreur colorimétrique. Cette dernière est composée d'une notion d'*importance visuelle* de chaque sommet et d'*erreur colorimétrique de contraction* de chaque arête. L'importance visuelle mesure l'écart en couleur d'un sommet vis-à-vis de ses voisins. Elle est additionnée lors des contractions : le sommet résultant a pour importance visuelle la somme de celle de ses prédécesseurs. L'erreur colorimétrique de contraction mesure l'écart entre deux scalaires : le premier étant la somme des écarts de couleur entre le sommet disparaissant v_0 et les couleurs de ses voisins ; la seconde étant l'écart de ces mêmes voisins à leur futur sommet voisin, à savoir v_1 .

Nous n'entrons pas dans les détails calculatoires ici car cela n'est pas particulièrement utile pour nous. Remarquons simplement qu'en cas de semi-contraction, il faut considérer deux contractions par arête car l'erreur est non symétrique : $\Delta((v_0, v_1) \rightarrow v_0) \neq \Delta((v_0, v_1) \rightarrow v_1)$.

2.3 Bilan de l'existant

Contraction d'arêtes. L'algorithme par contractions successives d'arêtes est devenu populaire. Il est standard dans de nombreuses bibliothèques de gestion de maillages, notamment *CGAL* [1996], *Multi-Tesselation* [1998], *OpenMesh* [Botsch *et al.*, 2002], *MeshLab* [2005], *CGoGN* et [Kraemer *et al.*, 2013]. Outre les avantages que nous avons listés dans la section 1, cet algorithme permet d'être étendu aux maillages progressifs voire au raffinement adaptatif.

Définition

Maillage progressif [Hoppe, 1996] : structure de données permettant de naviguer à travers les différentes simplifications d'un maillage. Il est construit par mémorisation, lors de la phase de simplification, de la suite d'arêtes contractées jusqu'à arriver au maillage le plus simplifié composé de quelques triangles seulement. Ce maillage grossier forme ensuite l'élément de base du maillage progressif. À partir de celui-ci, un maillage plus détaillé peut être obtenu par raffinements progressifs sous forme d'éclatements de sommets successifs, soit l'opération inverse de la contraction. Cette structure permet la visualisation et la transmission progressive.

Raffinement adaptatif [Hoppe, 1997] . Le raffinement adaptatif adapte le niveau de détail localement, en fonction des besoins de champ de vision. Une partie centrale à l'écran sera très détaillée alors que les parties périphériques ou non visibles seront plus grossières. Ceci fait intervenir des notions de point de vue et de visualisation. Le raffinement adaptatif permet de gagner en vitesse d'affichage.

En marge de l'algorithme de simplification, il faut faire attention aux contraintes topologiques que l'on souhaite considérer, notamment en termes de préservation des bords et en qualité du maillage triangulé (*e.g.* régularité des triangles, triangles dégénérés, valence des sommets) [Gourdon, 1995; Guézic, 1995]. Certaines arêtes pourront ainsi être pénalisées très fortement voire interdites de contraction. Il en va de même pour les arêtes du bord dans le cas de maillages ouverts. Ceci n'est pas important en vue de la définition d'un (nouveau) critère mais devra être pris en compte dans l'implémentation.

Métriques. Nous avons vu une métrique géométrique efficace et générant de bons résultats : la métrique d'erreur quadratique. Le critère d'efficacité est d'importance moindre pour nous, car il s'agit de pré-calcul : les différents niveaux de simplification sont pré-calculés pour être utilisés *a posteriori*. Nous avons vu qu'elle a été étendue efficacement à d'autres attributs scalaires ou vectoriels. Quelques métriques traitent de données en espace texture, de façon cumulée avec la QEM.

Extensibilité. À notre connaissance il n'y a cependant pas de métriques portant sur des attributs fonctionnels.

L'extension de métriques telles que la QEM à nos attributs fonctionnels ne peut se faire de façon triviale. La seule possibilité serait de considérer le vecteur de coefficients $[c_1 \cdots c_N]$ d'une fonction de radiance $R(\omega) = \sum_{i=1}^N c_i \phi_i(\omega)$ comme un attribut vectoriel afin d'appliquer une QEM de dimension $(3 + N)$ dessus.

Or, la distance entre coefficients de fonctions est nullement la distance entre fonctions de radiance : chaque coefficient étant un facteur d'un élément différent d'une base Φ , l'espace de dimension N considéré serait non homogène. De plus, si l'utilisateur

choisit d'exprimer les fonctions en un repère local, la comparaison de coefficients est un non-sens. Ce calcul serait possible en repère global. Mais là aussi, l'écart entre coefficients n'équivaut pas à l'écart entre valeurs de la fonction. Et un obstacle serait la gestion des ordres entre les différentes entités de la métrique.

Métrique combinée. Ceci nous amène à la problématique de la pondération entre entités mesurées : aucune des publications mentionnées n'explique comment le gérer. La pondération relative entre entités de différente nature est implicite dans la métrique QEM : cela dépend des différents ordres de grandeur. Ainsi, il est d'usage de normaliser tous les attributs : géométrie dans une boîte englobante de taille $[0, 1]^3$, idem pour la couleur, les normales, les coordonnées de textures, etc. Or, ceci ne garantit pas que les écarts entre différentes entités soient de même ordre : l'erreur géométrique grandit de façon quadratique en la taille du maillage, alors que les autres entités ne grandissent pas. Ceci implique que le poids relatif de la géométrie dans cette métrique varie tout au long de la simplification, ce qui est d'autant plus vrai avec la version cumulative de la QEM où les erreurs grandissent non seulement en fonction de la densité de maillage mais également en fonction de la quantité d'arêtes d'une arête à contracter.

Nous proposons d'établir une métrique cumulant une erreur géométrique (dans \mathbb{R}^3 et non dans un espace de plus haute dimension) avec une métrique propre aux informations colorimétriques.

$$\Delta = \mu E_{geom} + (1 - \mu) E_{colorimetric} \quad (6.3)$$

Afin de pleinement contrôler le poids relatif entre ces deux entités, nous utilisons deux mesures différentes pondérées par un coefficient. La géométrie est mesurée par la QEM non cumulative et nous veillons à ce que la métrique sur la couleur ait un ordre de grandeur similaire : elle doit grandir quadratiquement par rapport à la taille du maillage.

3 Nouveau critère combinant géométrie et couleur

Dans le cadre de l'algorithme de simplification par contraction d'arêtes, nous proposons une nouvelle métrique combinant un terme géométrique avec un terme sur la couleur. L'erreur géométrique est mesurée par la QEM non cumulative. L'erreur de couleur est définie dans cette section. Nous l'introduisons afin de l'étendre aux fonctions de radiance par la suite.

Nous ne considérons que la semi-contraction. Dans les résultats qualitatifs (section 3.3), nous montrons que l'impact de ce choix en termes de qualité est faible. Et dans la section 5.3, nous présentons ce qui rend la contraction optimale difficile à mettre en place. Nous traitons les couleurs attachés aux sommets du maillage et discutons de l'extension de nos travaux à une texture (section 5.2).

3.1 Critère sur la couleur

Nous disposons d'une surface visualisée dont la géométrie et la couleur sont définies de façon linéaire par morceaux. Cette surface est stockée en un ensemble discret d'éléments (*i.e.* les sommets) qui sont interpolés bi-linéairement. Le calcul d'erreur que nous proposons est fondé sur le maillage tel qu'il est rendu. Il mesure l'impact d'une semi-contraction d'arête sur la couleur en tout point du maillage, et non seulement aux sommets. Concrètement, il mesure la différence entre la couleur d'un sommet disparu (par la contraction) et celle qui apparaîtra en ce même point lorsque le maillage simplifié sera rendu.

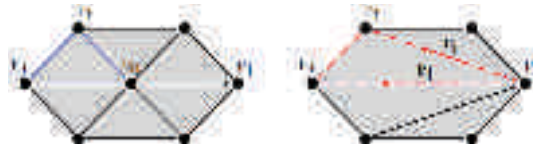


Figure 6.4 : Illustration d'une semi-contraction. L'arête (v_0, v_1) est contractée sur v_1 . Cette figure illustre la portion de surface dont la couleur est impactée et uniquement celle-ci. Deux sommets voisins consécutifs de v_0 sont notés v_i et v_j .

La figure 6.4 illustre une semi-contraction agrémenté des notations que nous utilisons. Soit \mathbf{p} la position et C la couleur d'un point. Nous définissons l'erreur colorimétrique de contraction par la formule suivante :

$$E_{col} = \sum_{\Delta_{ij1}} Aire(\Delta_{ij1}) d_{col}^2(C_0, C_1 + \nabla C_{ij1} \cdot (\mathbf{p}_0 - \mathbf{p}_1)) \quad (6.4)$$

L'explication de cette équation est la suivante.

- $\sum_{\Delta_{ij1}}$: l'erreur de contraction est une somme d'erreurs sur les triangles Δ_{ij1} après contraction.
- $Aire(\Delta_{ij1})$: l'erreur est proportionnelle à l'aire du triangle.
- d_{col}^2 : l'erreur sur chaque triangle est proportionnelle à une distance quadratique de couleurs. Cette distance représente le changement de couleur causé par la contraction. Ce changement de couleur se mesure entre couleurs à la position \mathbf{p}_0 avant et après contraction.
- C_0 : la couleur en \mathbf{p}_0 avant contraction.
- $C_1 + \nabla C_{ij1} \cdot (\mathbf{p}_0 - \mathbf{p}_1)$: la couleur en \mathbf{p}_0 après contraction. Il s'agit du résultat de l'interpolation bi-linéaire sur le triangle Δ_{ij1} ⁸ car ∇C_{ij1} est le gradient de couleur sur Δ_{ij1} .

Avant d'expliquer comment calculer le gradient sur un triangle, nous souhaitons montrer quelques illustrations soutenant cette métrique. La figure 6.5 nous montre des cas extrêmes, avec des dégradés entre le noir et le blanc. Les exemples (a) et (b) montrent des glissements dûs à la contraction qui sont colinéaires au dégradé

8. Notons que cette formule est correcte du fait que le gradient soit constant sur le triangle grâce à l'interpolation bi-linéaire des couleurs aux sommets.

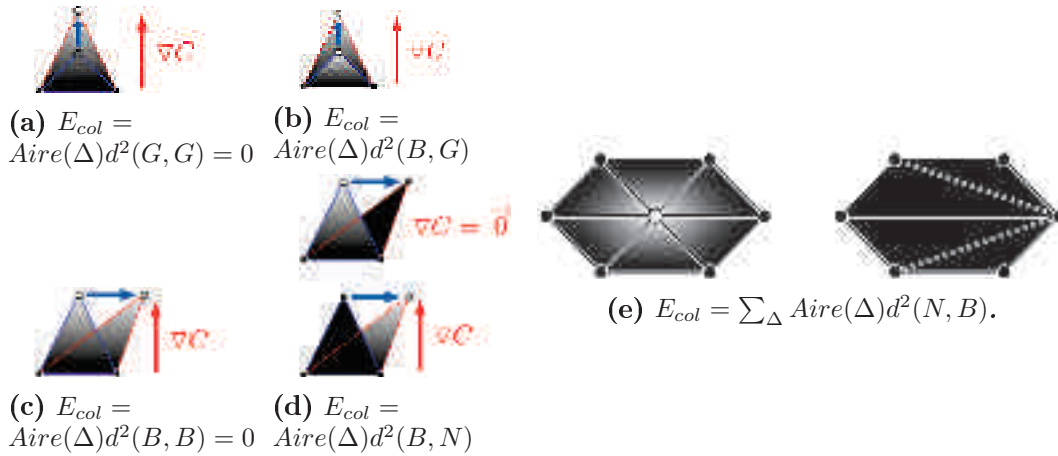


Figure 6.5 : Illustrations de calculs de E_{col} sur des exemples synthétiques où des couleurs extrêmes (N : noir, G : gris, B : blanc) sont interpolées. Exemples (a) à (d) : calcul de E_{col} par triangle. La direction du gradient de couleur est donnée en rouge. Exemple (e) : calcul de E_{col} pour une semi-contraction complète faisant disparaître un sommet blanc entouré de noirs. L'erreur E_{col} résultante est donnée sous chaque figure.

de couleur⁹. Dans (a), le gradient de couleur est respecté et notre métrique mesure $E_{col} = 0$. Dans (b), le gradient de couleur n'est pas respecté : notre métrique mesure $E_{col} = Aire \times d(B, G)$, soit l'aire fois la distance colorimétrique au point d'intérêt (*i.e.* blanc-gris).

Les exemples (c) et (d) montrent des déplacements de contraction orthogonaux au gradient de couleur. Pour (c), l'erreur mesurée est nulle car la couleur ne varie pas. Pour (d), la couleur varie et l'erreur mesurée est en proportion de cette variation. L'erreur complète d'une contraction dans le cas de la disparition totale d'une couleur (figure 6.5e) vaut en conséquence cet écart de couleur fois l'aire impactée par l'interpolation.

Avantages. La métrique E_{col} mesure correctement différents phénomènes locaux pouvant se produire sur la couleur tels l'étirement ou la disparition de couleurs. Le calcul n'est pas bien complexe et aucune paramétrage locale n'est nécessaire.

Grâce au fait que la métrique soit basée sur le gradient, il y a un certain *équilibre* entre les mesures de géométrie et de couleur. Cet équilibre est invariant à la mise à l'échelle et au niveau de simplification car :

- l'erreur couleur grandit quadratiquement par rapport à la taille du maillage, et
- la QEM en fait de même lorsqu'on considère la version non cumulative.

C'est pourquoi nous avons choisi cette dernière.

Aussi, la métrique s'étend presque trivialement aux attributs fonctionnels représentant de la couleur, ce qui est notre objectif. Finalement, les résultats de la simplifica-

9. Par « glissement » on entend le glissement du sommet v_0 sur v_1 .

tion avec cette métrique rivalisent avec les standards actuels : notamment la métrique QEM étendue à la couleur (voir section 3.3).

3.2 Calcul du gradient

Le calcul du gradient de couleur sur la surface est nécessaire dans l'équation (6.4). La couleur rendue est linéaire par morceaux, ce qui implique que le gradient est constant sur le triangle Δ_{ij1} . Soit $\mathbf{e}_i = (v_1, v_i)$ et $\mathbf{e}_j = (v_1, v_j)$ (voir figure 6.4), alors la différentiation sur la surface nous donne¹⁰ :

$$\nabla C_{ij1} = J(J^T J)^{-1} \begin{bmatrix} C_i - C_1 \\ C_j - C_1 \end{bmatrix} \text{ où } J = \begin{bmatrix} \mathbf{e}_i & \mathbf{e}_j \end{bmatrix} \quad (6.5)$$

Ainsi, ∇C_{ij1} vaut

$$\frac{1}{(2\text{Aire}(\Delta_{ij1}))^2} \left((\|\mathbf{e}_j\|^2(C_i - C_1) - \mathbf{e}_i \cdot \mathbf{e}_j(C_j - C_1)) \mathbf{e}_i + (\|\mathbf{e}_i\|^2(C_j - C_1) - \mathbf{e}_i \cdot \mathbf{e}_j(C_i - C_1)) \mathbf{e}_j \right) \quad (6.6)$$

Pour rendre le tout simple et calculable, nous avons fait deux approximations.

- En toute rigueur, C est une couleur ($C \in \mathbb{R}^3$) et ∇C devrait être une matrice Jacobienne. Nous avons cependant considéré ici que C est la valeur scalaire d'un des canaux, tous traités séparément. Ainsi, ∇C est bien un vecteur. Nous avons empiriquement constaté qu'utiliser un espace de couleurs perceptuellement uniforme n'apporte que peu. Ainsi nous utilisons l'espace RVB dont les canaux sont homogènes.
- Les gradients calculés sont valables à l'intérieur des plans des triangles. La formule (6.4) calcule ainsi des interpolations valides dans le triangle uniquement. Or, p_0 ne se situe dans le cas général pas dans le triangle, ni même dans son plan. Dans la formule 6.5, nous avons implicitement étendu la définition de la couleur rendue à \mathbb{R}^3 : la couleur est linéaire dans le plan et constante dans la direction orthogonale.

3.3 Résultats qualitatifs

Le but de ce critère colorimétrique est non pas de surpasser les standards actuels, mais de produire des résultats convaincants et d'être extensible à la radiance. Nous montrons ici que la simplification utilisant notre critère rivalise avec les standards actuels. Nous allons nous comparer à la méthode QEM étendue d'un attribut de couleur [Hoppe, 1999] que l'on placera dans les mêmes conditions que la nôtres :

10. Voir annexe B.

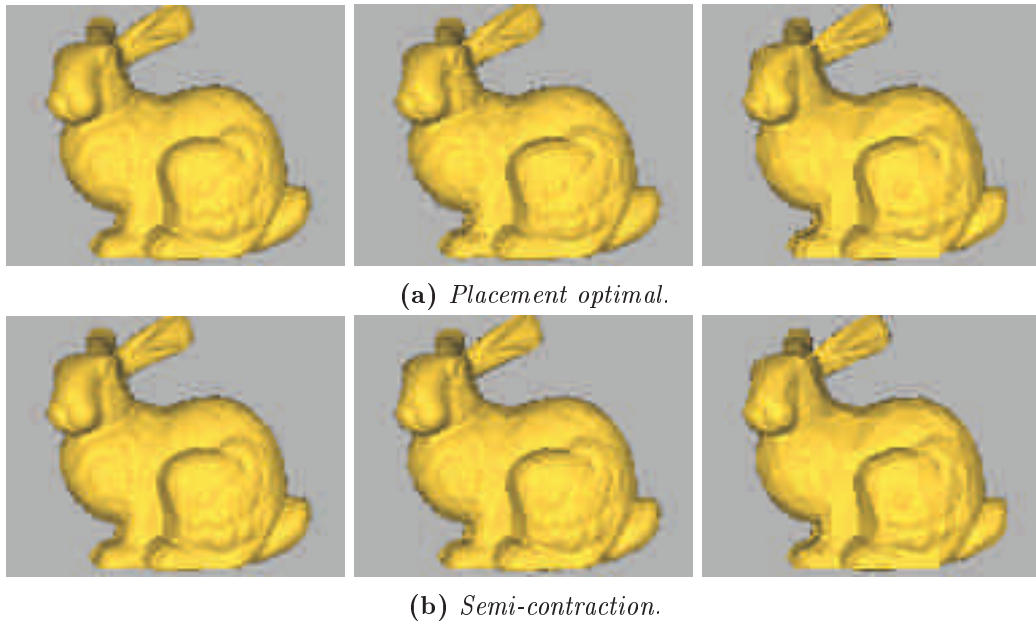


Figure 6.6 : *Comparaison entre simplifications par contraction avec placement optimal ou semi-contractions, à l'aide de la QEM non cumulative. Le modèle original (à gauche) a 66 562 sommets. Les modèles simplifiés n'en ont plus que 2 080 (milieu) et 520 (à droite). Il est difficile de qualifier un résultat de meilleur que les autres.*

semi-contraction et non cumul des erreurs.

La QEM avec placement optimal génère cependant de meilleurs résultats que la seule semi-contraction. Elle génère en effet des triangles plus homogènes et plus proches de la forme équilatérale. En pratique, la différence ne se voit que pour de forts taux de simplification : le taux de simplification à partir duquel le modèle est jugé « trop dégradé » est légèrement supérieur si l'on exploite le placement optimal. La différence sera minime particulièrement pour des maillages numérisés car ceux-ci sont peu homogènes dès leur construction en comparaison avec les maillages modélisés. La figure 6.6 illustre la différence quasiment imperceptible entre ces deux variantes.

La figure 6.7 compare la QEM étendue (contraction optimale et semi-contraction) et notre métrique sur une simplification du modèle du masque comportant 193 000 sommets sur lesquels une couleur diffuse est définie¹¹. On y voit que toutes les métriques préservent bien les caractéristiques colorimétriques prononcées (notamment sur le front). Les veinules de bois du masque sont encore visibles au premier taux (25% de sommets restants) alors qu'au second (1,5%) seul les caractéristiques colorimétriques principales sont préservées. La différence entre les trois métriques est faible.

11. Cette couleur a été définie par l'extraction d'une couleur moyenne à partir des fonctions de radiance acquises.

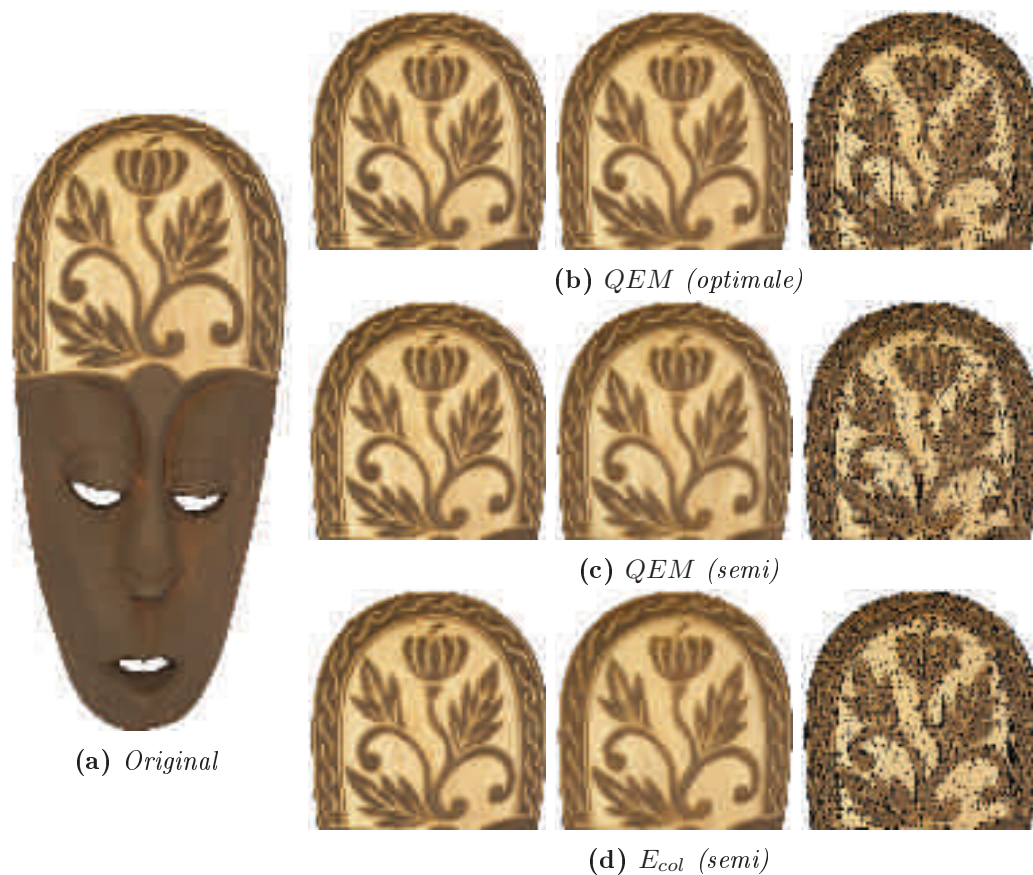


Figure 6.7 : Simplification du masque avec trois critères de couleur. Le modèle original contient 192 757 sommets. La première simplification (à gauche) ne contient plus que 48 189 soit 25% des sommets. La seconde (au centre) en contient 3 011, soit 1,5%. La dernière colonne illustre ce même modèle avec le maillage en fil-de-fer sur-imprimé.

Bilan. La métrique que nous venons de proposer fonctionne bien : il n’y a pas de différence nette entre E_{col} et E_{QEM} . Elle intègre la façon dont le maillage coloré est rendu : avec une interpolation bi-linéaire des valeurs aux sommets. De plus, elle n’est pas coûteuse en calculs : *i.e.* il s’agit principalement d’évaluer la formule (6.5). Et surtout, contrairement aux autres méthodes existantes, elle est extensible aux attributs de radiance grâce à l’interpolation de fonctions de radiance que nous avons définis dans le chapitre 5. Nous voyons l’importance de l’établissement d’un critère sur la radiance dans les résultats de la section suivante.

4 Critère combinant géométrie et radiance

Nous mettons en place une métrique sur un champ de lumière surfacique représenté par des fonctions de radiance. Elle est combinée d’après l’équation 6.3 avec la mé-

trique géométrique. En chaque sommet, nous disposons d'une fonction $R(\omega)$ et d'un repère indiquant l'orientation de l'hémisphère visible (si la fonction est définie dans un repère global (et donc sphérique), la normale n indiquant la moitié visible de la sphère suffit).

Nous intégrons l'écart de couleur défini dans la section 3 sur l'hémisphère afin de définir un écart sur la radiance. Il s'agit principalement de remplacer l'écart de couleur par un écart entre fonctions dans l'équation (6.4). On souhaite que cet écart prenne en compte le maillage rendu, qui est défini continu par morceaux grâce à l'interpolation de la géométrie et de la radiance tel qu'établi au chapitre précédent. Il prendra donc en compte la façon dont la radiance est rendue en tout point : par interpolation de la fonction symétrisée.

4.1 Critère sur la radiance

La formule générale de la métrique est la suivante :

$$E_{RF} = \sum_{\Delta_{ij1}} \text{Aire}(\Delta_{ij1}) \left\| \tilde{R}_0 - \tilde{R}_1 - \nabla \tilde{R}_{ij1} \cdot (\mathbf{p}_0 - \mathbf{p}_1) \right\|_{L_2}^2 \quad (6.7)$$

où la norme L_2 sur les fonctions de carré intégrable sur l'hémisphère \mathcal{H} est utilisé :

$$\|R\|_{L_2} = \left(\frac{1}{2\pi} \int_{\omega \in \mathcal{H}} R(\omega)^2 d\omega \right)^{1/2} \quad (6.8)$$

Cette métrique est analogue à celle de l'équation (6.4). Elle met en œuvre la distance suivante entre fonctions de radiance :

$$d_{RF}(R_1, R_2) = \left\| \tilde{R}_1 - \tilde{R}_2 \right\|_{L_2} \quad (6.9)$$

Les différents choix sont justifiés :

Représentation symétrisée. La métrique prend en compte la radiance interpolée en tout point de la surface selon la formule définie au chapitre précédent. Pour ce faire, nous utilisons exclusivement les représentations symétrisées \tilde{R} . Ceci permettra de déterminer que deux fonctions sont différentes si et seulement si cette différence n'est pas due à un changement de matériau ou d'illumination. En effet, si l'on s'autorise une contraction de l'arête reliant deux sommets voisins dont la radiance symétrisée est identique ou presque, la visualisation interpolée en ce point (D'après la définition de l'équation (5.2) de la page 125.) sera identique malgré la contraction.

Notons par ailleurs qu'alors que $\tilde{R}_1 - \tilde{R}_2$ n'est pas le symétrisé de $R_1 - R_2$ (leurs normales divergent), on a bien $\|R\|_{L_2} = \|\tilde{R}\|_{L_2}$ et R et \tilde{R} sont définies sur le même hémisphère.

Norme utilisée. nous utilisons la norme L_2 sur les fonctions de carré intégrable par cohérence avec la métrique couleur. On souhaite en effet que la distance entre deux fonctions de radiance diffuses soit égale à celle entre les couleurs : *i.e.* $E_{RF} = E_{col}$ si les fonctions de radiance sont constantes.

Hémisphère d'intégration. D'après l'équation (6.9), la distance entre fonctions de radiance est définie sur un hémisphère. Or, deux fonctions sont généralement définies sur des hémisphères différents : $\mathcal{H}_0 \neq \mathcal{H}_1$. Nous devons donc choisir *a priori* sur quel hémisphère évaluer l'équation (6.7). Par cohérence avec la normale interpolée au point d'évaluation \mathbf{p}_0 , nous choisissons $\mathcal{H} = \mathcal{H}_0$.

Lorsque $\mathcal{H}_1 \neq \mathcal{H}_0$, l'intégrale nécessite d'évaluer \widetilde{R}_1 hors de son hémisphère car $\mathcal{H}_1 \not\subseteq \mathcal{H}_0$. Pour ce faire, nous prolongeons la définition de la fonction par une constante sur l'hémisphère opposé, à savoir la couleur moyenne calculée sur l'hémisphère de définition. Ce choix est fait par souci de cohérence avec la métrique couleur.

4.2 Calcul de l'erreur

Gradient. Le gradient $\nabla \widetilde{R}_{ij1}(\omega)$ des fonctions symétrisées est une fonction sur l'hémisphère. Il peut être défini en remplaçant C par \widetilde{R} dans l'équation (6.5), soit :

$$\begin{aligned} \nabla \widetilde{R}_{ij1}(\omega) = \frac{1}{(2Aire(\Delta_{ij1}))^2} & \left(\left(\|\mathbf{e}_j\|^2 (\widetilde{R}_i(\omega) - \widetilde{R}_1(\omega)) - \mathbf{e}_i \cdot \mathbf{e}_j (\widetilde{R}_j(\omega) - \widetilde{R}_1(\omega)) \right) \mathbf{e}_i \right. \\ & \left. + \left(\|\mathbf{e}_i\|^2 (\widetilde{R}_j(\omega) - \widetilde{R}_1(\omega)) - \mathbf{e}_i \cdot \mathbf{e}_j (\widetilde{R}_i(\omega) - \widetilde{R}_1(\omega)) \right) \mathbf{e}_j \right) \end{aligned} \quad (6.10)$$

Cette formule est par la suite intégrée sur $\omega \in \mathcal{H}$ selon l'hémisphère choisi.

Intégration. Le calcul de l'erreur (équation (6.7)) requiert l'intégration du mélange de fonctions hémisphériques (voir équation (6.8)). Chacun de ses termes étant une combinaison linéaire de fonctions, définir une intégrale analytique s'avère complexe voire impossible du fait des nécessaires ré-alignements des repères locaux. Ainsi, nous évaluons cette intégrale numériquement en utilisant une quadrature de *Lebedev* [1976] sur la sphère.

4.3 Résultats qualitatifs

La métrique locale utilisée pour la simplification de champs de lumière est la première en son genre. Il n'est ainsi pas possible de faire une myriade de comparaisons. Nous avons toutefois mis en exergue le besoin d'un critère sur la radiance. Pour cela, nous avons procédé à des simplifications avec des critères *ad-hoc* composés de métriques existantes :

- Critère purement géométrique (QEM). Ceci garantit une bonne approximation géométrique, mais les caractéristiques colorimétriques ne sont pas prises en compte.
- Critère géométrique + couleur (QEM étendue et E_{col}) en prenant pour couleur la couleur moyenne de chaque fonction de radiance. La couleur est prise en compte, les caractéristiques directionnelles non.
- Critère géométrique + couleur + normale (QEM étendue) : ainsi, tant la couleur que les effets liés au changement de normale seront pris en compte.

Seule la semi-contraction peut être considérée afin de garantir la possibilité de visualiser la radiance sur les maillages simplifiés (la radiance est toujours présente sur les sommets simplifiés). Nous estimons qu'il s'agit là des critères les plus adaptés aux champs de lumière utilisant des outils existants.

À côté de cela, nous avons notre simplification dont la métrique est une combinaison linéaire du critère géométrique (QEM) et de E_{RF} . En pratique, nous avons normalisé nos modèles (les coordonnées géométriques sont alors comprises entre 0 et 1). Nous utilisons la ratio suivant : 1% d'erreur géométrique et 99% d'erreur sur la radiance ($\mu = 0,01$ dans l'équation (6.3)). Ce ratio a été défini empiriquement. Il est aisé de le définir grâce à l'indépendance de nos deux critères et leur homogénéité de leur proportion à la taille du maillage. On peut remarquer que ceci n'est pas le cas pour les métriques de QEM étendues.

4.3.1 Données synthétiques

La figure 6.8 illustre un cas synthétique que nous avons mis en place spécifiquement pour montrer le besoin d'une mesure sur la radiance. Un modèle sphérique composé de triangles homogènes contient des fonctions de radiance placées en ses sommets. Celles-ci encodent un modèle de type *Phong* avec illumination par deux sources lumineuses ponctuelles éloignées. Deux bandes de sommets ne contiennent que des radiances diffuses (*i.e.* fonctions constantes). Les deux demi-sphères ont également des couleurs diffuses différentes afin d'illustrer tant des changements en termes de caractéristiques de couleur et de caractéristiques directionnelles.

Le critère géométrique permet de bien simplifier la sphère géométriquement : le maillage qui en résulte est relativement régulier. Ni les couleurs, ni les caractéristiques directionnelles ne sont cependant prises en compte.

Les critères colorimétriques permettent de bien préserver la ligne de triangles séparant les couleurs principales (bleu et rouge), mais ne fait pas la différence entre bandes non spéculaires et spéculaires.

Le critère de normale va plutôt plaider pour des courbures uniformément échantillonnées (en termes de sommets, donc de tailles de triangle) : sur la sphère, ce critère cherche à simplifier les zones proches des pôles, beaucoup plus denses. Elle ne règle en aucun cas le problème de détection d'erreurs entre zones spéculaires et bandes non spéculaires.

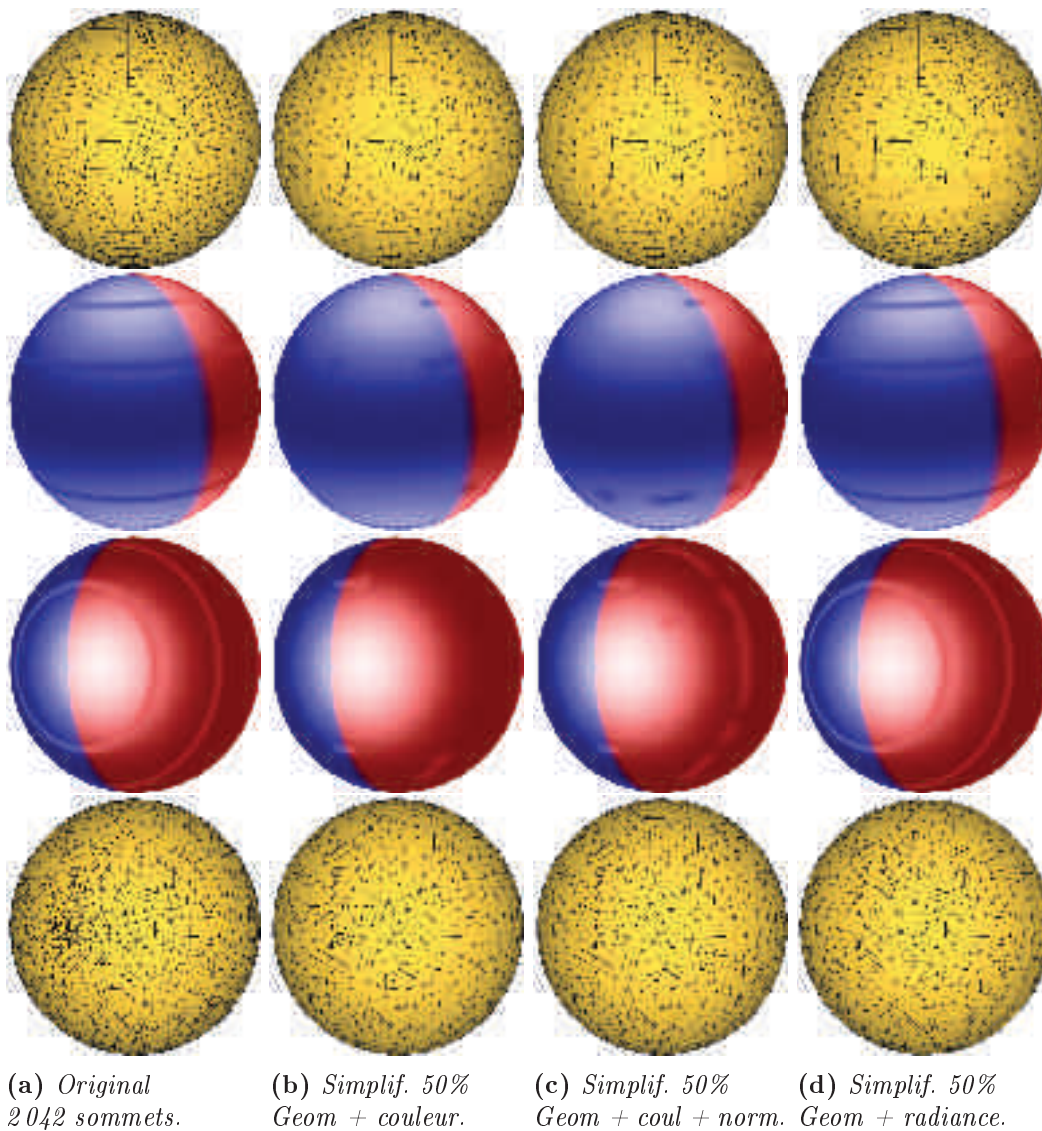


Figure 6.8 : *Simplifications avec trois critères différents d'un modèle avec radiance synthétique et visualisation du résultat depuis deux points de vue. 6.8b : critère combinant géométrie et couleur. 6.8c : critère combinant géométrie, couleur et orientation locale (normales). 6.8d : critère combinant géométrie et radiance. Les critères de couleur exploitent la couleur moyenne (intégration sur l'hémisphère) de chaque fonction de radiance.*

Enfin, le critère mesurant un écart sur la radiance permet de bien préserver l'aspect global, prenant en compte de façon équilibrée la géométrie, la couleur et les caractéristiques directionnelles.

4.3.2 Modèles numérisés

Les différences sont moins flagrantes sur nos données numérisées mais nous pouvons tout de même les analyser et remarquer les avantages du critère sur la radiance. La figure 6.9 illustre le modèle du masque.

La partie supérieure (*i.e.* le front) contient des motifs de couleurs nets. La simplification avec E_{RF} génère des résultats comparables – en termes de préservation de motifs – à la simplification avec critère sur la seule couleur. En ce sens, elle est satisfaisante.

La partie inférieure (*i.e.* le visage) contient des caractéristiques directionnelles fortes. On peut voir sur la figure que les reflets ne sont bien préservés sur les maillages simplifiés que lorsque le critère sur la radiance a été utilisée : par exemple les reflets sur le nez ou autour des sourcils. Le maillage est plus dense là où la radiance exhibe des caractéristiques directionnelles changeantes.

Afin de bien apprécier les capacités de cet algorithme et du critère choisi, nous montrons quelques simplifications successives du *double dragon*. Pour rappel, ce dernier exhibe des caractéristiques fortes à la fois en termes de géométrie, de couleur et de reflets directionnels. La figure 6.10 nous montre que les modèles simplifiés approchent bien le modèle original jusqu'à des taux de simplification drastiques. Les dernières simplifications montrent des lacunes en termes de couleur (flou). Ces modèles peuvent cependant être satisfaisants pour une visualisation à distance ou sur de petits écrans.

5 Conclusion

Nous avons proposé une nouvelle méthode de simplification intégrant un critère sur la radiance en plus d'un critère géométrique. Cette contribution est la première à proposer une métrique de simplification basée sur des critères directionnels ou fonctionnels. Ainsi, nous avons proposé une comparaison à des outils existants adaptés « au mieux » pour gérer ces attributs. Nous avons montré qu'il y a un réel besoin de prendre en compte ces paramètres directionnels (figures 6.8 et 6.9) : ceci est flagrant sur les cas synthétiques et détectable sur les données numérisées. Nos tests montrent ensuite que les simplifications successives sont de qualité raisonnable, générant des maillages trouvant un bon compromis entre complexité et réalisme.

5.1 Bilan

Dans son article [1996], *Hugues Hoppe* commence par identifier les problématiques liées aux maillages obtenus par scans 3D. Il liste plus particulièrement les besoins suivants :



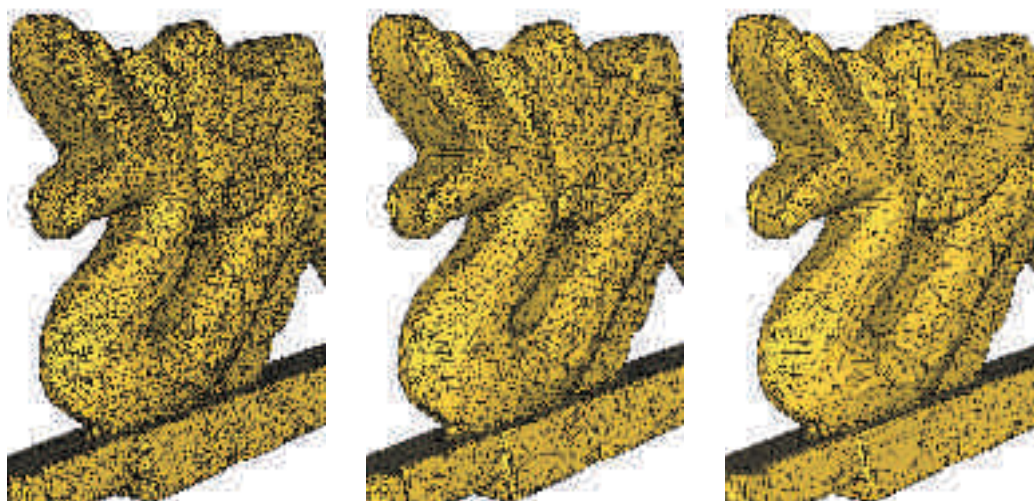
Figure 6.9 : Comparaison de simplifications avec et sans critère sur les fonctions de radiance.

1. *Simplification de maillages* afin de visualiser en temps réel,
2. *Approximation à différents niveaux de détails* afin de n'afficher de la haute résolution que si cela est nécessaire,
3. *Compression de maillages* pour un niveau de détail donné, afin de limiter le coût de stockage,
4. *Transmission progressive* afin de garantir une qualité visuelle progressive lors de la transmission de données en affichant d'abord un maillage grossier puis plus de détails lors de l'arrivée des données,
5. *Raffinement adaptatif* afin d'adapter le niveau de détail en fonction des régions du maillage visualisés.

Il propose ensuite successivement ses outils de simplification par contraction d'arêtes, de maillages progressifs puis de raffinement adaptatif [Hoppe, 1997].



(a) Pleine résolution : 921 218 sommets.



(b) Simplifié (6%) :
57 576 sommets.

(c) Simplifié (3%) :
28 788 sommets.

(d) Simplifié (1,5%) :
14 394 sommets.

Figure 6.10 : Simplifications successives du double dragon.

Le maillage progressif permet la transmission progressive grâce à la définition d'un maillage grossier suivi d'une séquence d'éclatements de sommets.

Le raffinement adaptatif permet de gérer différents niveaux de détails dans différentes parties du maillage en fonction de la vue courante. Pour cela, la séquence de contractions d'arêtes (respectivement d'éclatements de sommets) ne doit plus être linéaire mais structurée sous forme d'arbre, afin de pouvoir affiner une branche alors qu'une autre reste grossière.

Notre contribution initie une réponse à ces besoins pour des maillages avec radiance et issus de la numérisation. Dans ce manuscrit, nous avons directement répondu aux trois premiers points.

Pour répondre aux besoins suivants, une possibilité est de mettre en place un maillage progressif, puis le raffinement adaptatif. La plupart des obstacles restants sont liés à la topologie et à la structure de la suite d'arête à contracter/à éclater et non à la complexité des attributs : tout maillage progressif peut être adapté pour le raffinement adaptatif. Pour en créer un il suffirait de stocker exhaustivement toutes les contractions ainsi que leur ordre pour définir un maillage progressif. Celui-ci pourra être améliorée dans un second temps en intégrant un mécanisme de prédiction afin de permettre une meilleure compression.

5.2 Perspectives

Évaluation quantitative. Nous nous sommes limités à une évaluation qualitative de nos résultats. Dans le chapitre 4, nous avons cependant affirmé notre attachement à la création d'évaluations numériques. Nous aimerions ainsi mettre en place une bonne métrique d'écart entre maillages (non forcément pourvus de fonctions de radiance) que l'on souhaiterait corrélée à la perception humaine du rendu de ce dernier. Une métrique entre maillages rendus est complexe à caractériser car elle varie en tout point de vue : elle dépend de sa radiance, stockée ou simulée.

C'est pour cette raison que nous souhaitons mettre en place une métrique d'écart entre maillages avec attribut de radiance et qui soit corrélée à la perception humaine. Ce n'est que récemment que les mesures classiques ont été évaluées par rapport à la perception que les utilisateurs en ont [Guthe *et al.*, 2009; Čadík *et al.*, 2012; Herzog *et al.*, 2012]. Déterminer les caractéristiques saillantes aux yeux d'un observateur est un problème ardu : des problématiques de vision et de perception subjective sont à prendre en compte, ce qui pose beaucoup de problèmes [Corsini *et al.*, 2013]. Il le devient davantage lorsque l'on considère un attribut de radiance.

Textures. Le mécanisme de simplification traite uniquement les maillages avec radiance attaché aux sommets. Si on place ces radiances dans une texture, la résolution géométrique (*i.e.* la tessellation) et la résolution de la radiance (*i.e.* la quantité de texels) sont dissociés. Ainsi la géométrie et la radiance pourraient être simplifiées de

façon séparées.

Une telle décorrélation est délicate car la géométrie et la radiance sont liées, cette dernière découlant notamment de la précédente, ainsi que de l'irradiance et du matériau composant la surface. Il est donc important qu'une quelconque métrique de simplification, même séparée, implique les deux entités. Certaines métriques proposent de mesurer l'erreur géométrique en utilisant la QEM, puis de pénaliser plus fortement certaines arêtes correspondant à des propriétés importantes en espace texture ou en pixels-écran [Cohen *et al.*, 1998; Sander *et al.*, 2001; González *et al.*, 2007]. Par exemple, González *et al.* pénalise davantage les arêtes qui correspondent à un haut gradient de couleur dans l'image de texture 2D et Sander *et al.* minimisent l'étirement de la texture en contrôlant la paramétrage du maillage simplifié. Finalement, elles placent le sommet par minimisation de l'erreur géométrique et lorsque leur métrique le permet, elles opèrent une minimisation de l'erreur sur les attributs ensuite, étant donné la position géométrique.

Au chapitre 5, nous avons déjà mentionnés les difficultés de la définition d'une texture de fonctions de radiance : que ce soit pour l'interpolation ou la définition d'une *mip-map*, l'interpolation analytique des fonctions pose souci lorsqu'elles sont définies en repère local. Le *mip-mapping* peut former une solution adéquate pour représenter la radiance à différents niveaux de détails. Elle pourrait s'utiliser en complément d'un maillage progressif déterminé avec un critère géométrique et un critère sur la radiance, par exemple comme dans l'article [González *et al.*, 2007].

5.3 Discussion : contraction avec plongement optimal

Dans [Vanhoey *et al.*, 2010], nous avons proposé un outil de simplification de maillages avec radiance qui intègre bien une contraction avec plongement optimal. Pour ce faire, le critère utilisé ne considère que l'écart entre les radiances attachés aux sommets v_0 et v_1 . Une expression de cette erreur sous forme analytique et quadratique en les coefficients de fonction permet alors la définition d'un plongement optimal via la résolution d'un système linéaire.

Nous estimons désormais que cette métrique est *insuffisante* : elle est mise en défaut sur des cas particuliers. Dans la figure 6.11, on voit que deux sommets ont une radiance (constante) très proche. L'arête les reliant est donc prioritairement contractée d'après ce critère. La minimisation de l'écart à ces deux radiances génère une nouvelle radiance constante de même valeur et est incapable de mesurer des glissements ou étirements de couleur.

Ce constat nous a amené à établir une métrique qui prend en compte le rendu interpolé en tout point de la surface. Celle-ci ne nous permet cependant pas de définir une contraction optimale. Cela nécessiterait de pouvoir trouver la combinaison position-radiance qui minimise l'écart au modèle avant contraction. D'après la formule (6.7),

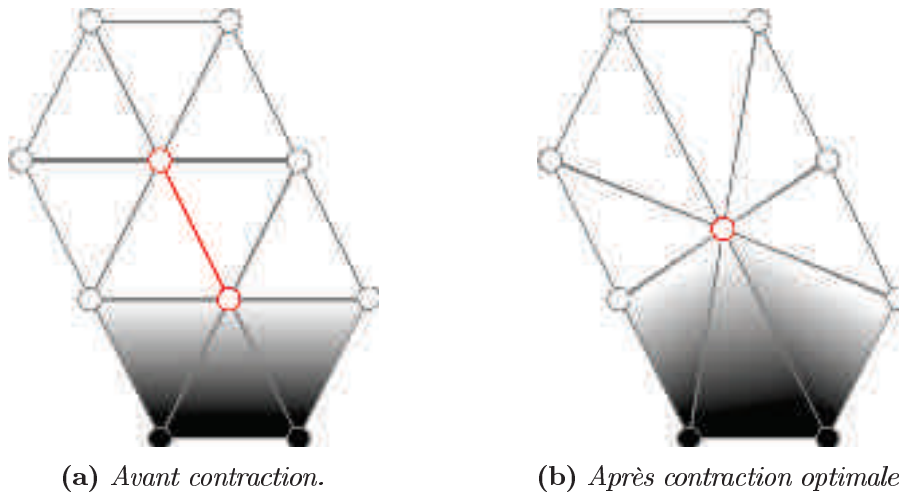


Figure 6.11 : Métrique d'erreur de radiance quadratique. L'écart entre les sommets incidents à l'arête rouge est nulle et celle-ci est donc contractée. Le nouveau sommet minimise l'écart à ces sommets. La métrique ne prends pas en compte l'étirement qui résulte de l'interpolation avec les voisins du nouveau sommet.

cet écart implique un certain nombre de fonctions de radiance.

Dans le cas d'un repère unique (*i.e.* global), un calcul analytique est possible : il est possible d'interpoler, moyenner et minimiser des radiances. Dans le cas contraire (*i.e.* repère local), il est difficile d'exprimer toutes les fonctions de la formule (6.7) dans un même repère. En base polynomiale paramétrée sur le plan projeté, ce calcul est même impossible.

TROISIÈME PARTIE

CONCLUSION

Chapitre 7

Conclusion

« Now this is not the end. It is not even the beginning of the end. But it is – I hope –, the end of the beginning ».

Détournement de la célèbre citation de Winston Churchill.

RÉSUMÉ:

Dans ce dernier chapitre, nous concluons par un résumé et une analyse de nos contributions dans un cadre plus large : celui du rendu basé image. Nous fournissons également quelques éléments empiriques permettant de guider le choix d'une bonne base de fonctions de radiance. Dans la section 3, nous présentons plusieurs perspectives. D'abord nous résumons celles évoquées dans les trois chapitres présentant nos contributions. Ensuite, nous discutons de la mise en place d'une étude comparative des représentations de fonctions de radiance, du rattachement de fonctions de radiance aux faces d'un maillage et de la reconstruction simultanée du maillage et de la radiance. Finalement, nous évoquons en perspective à plus long terme, le passage de nos contributions aux données fonctionnelles de plus haute dimension, par exemple les BTF.

Sommaire

1	Apports pour le rendu de champs de lumière	169
1.1	Rappels des objectifs	169
1.2	Nos contributions	170
2	Choix du modèle de radiance	173
3	Perspectives	175
3.1	Résumé des perspectives évoquées	175
3.2	Étude sur le choix des fonctions de radiance	176
3.3	Radiance par face	177
3.4	Reconstruction conjointe	179
3.5	Long terme	179

1 Apports pour le rendu de champs de lumière

Nous avons dans cette thèse débuté par placer le cadre puis décrire le domaine du rendu basé image dans lequel se situe la reconstruction du champ de lumière numérisé. Son objectif est l'échantillonnage par acquisition, la reconstruction, le stockage et la visualisation du champ de lumière. Nous avons dans ce cadre proposé un certain nombre de contributions appliquées aux champs de lumière surfaciques.

1.1 Rappels des objectifs

Nous souhaitons mettre nos contributions en perspective d'objectifs formulés il y a plus de dix ans dans l'article *Unstructured Lumigraph Rendering* [Buehler *et al.*, 2001]. Cet article fournit une liste intéressante de huit objectifs précis auxquels doit répondre un mécanisme de rendu de champ de lumière :

1. Utilisation d'une représentation géométrique : permet d'éviter les erreurs de projection et/ou de prendre en compte la parallaxe.
2. Acquisition non structurée : une reconstruction continue dans le domaine angulaire doit pouvoir être créée sans avoir recours à de grandes structures rigides d'acquisition ni de pré-détermination des positions d'acquisition.
3. Cohérence épi-polaire : un rayon acquis et traversant le centre de projection de la caméra virtuelle doit être favorisé par rapport aux autres. Ceci n'est pas le cas dans de nombreuses méthodes qui interpolent un sous-ensemble de rayons sur des critères de distance ou d'angle.
4. Déviation angulaire minimale : il faut favoriser le rayon acquis ayant l'angle le plus proche de celui du rayon virtuel. Ceci n'est pas forcément vrai, notamment pour le paramétrage classique du champ de lumière plénoptique [Levoy et Hanrahan, 1996; Gortler *et al.*, 1996].
5. Continuité : un déplacement infinitésimal de la caméra virtuelle doit générer une différence d'aspect suffisamment petite, notamment pour éviter les effets de *popping*.
6. Sensibilité à la résolution : s'il y a différentes acquisitions le long d'un même rayon, il faut rendre l'acquisition la plus proche de l'emplacement de la caméra virtuelle.
7. Cohérence des rayons équivalents : le long d'une ligne de vue, la même acquisition doit être visualisée en tout temps (sauf si stipulé autrement par la propriété numéro 6).
8. Visualisation temps-réelle.

L'article rapporte un tableau comparatif des techniques existantes à l'époque : [Levoy et Hanrahan, 1996; Gortler *et al.*, 1996; Debevec, 1996; Pulli *et al.*, 1997; Debevec *et al.*, 1998; Pighin *et al.*, 1998; Heigl *et al.*, 1999; Wood *et al.*, 2000; Buehler *et al.*, 2001] (voir tableau 7.1).

Prop	LH96	Go96	De96	Pu97	De98	Pi98	He99	Wo00	Bu01
1	-	+	+	+	+	+	+	+	+
2	-	R*	+	+	+	+	+	R*	+
3	+	+	+	-	-	-	+	+	+
4	-	-	+	-	+	+	-	+	+
5	+	+	+	+	-	+	+	+	+
6	-	-	-	-	-	-	-	-	+
7	+	+	+	+	+	+	-	+	+
8	+	+	-	+	+	+	+	+	+

*au prix d'un ré-échantillonnage.

Tableau 7.1 – Propriétés du rendu champ de lumière d'après [Buehler *et al.*, 2001].

En 2001, l'exploitation des données géométriques pour les champs de lumière n'en était qu'à ses balbutiements. Seul [Wood *et al.*, 2000] permet de représenter une géométrie détaillée et de centrer les données sur la surface : c'est le premier véritable champ de lumière surfacique. Même si Buehler *et al.* répondent à tous les critères qu'ils ont énoncés, il n'exploitent qu'une approximation grossière de la géométrie. De nombreuses propriétés sont respectées grâce à un calcul complexe de pondération de rayons à interpoler. De plus, les bons résultats reposent sur du sur-échantillonnage photographique dans le domaine angulaire.

Nous avons choisi d'utiliser la représentation du champ de lumière *surfactive* avec radiance définie *localement* sur le maillage. Ce choix est motivé par la finesse de la représentation géométrique et l'évitement des problèmes de parallaxe de la représentation plénoptique. Il implique de fait le respect des propriétés 1 (utilisation de la géométrie), 3 à 5 (le regroupement des données par rayon émis évite les erreurs de sélection du bon rayon) et 7 (ce même regroupement définit l'unicité d'un rayon sortant). Le critère 6 (sensibilité à la résolution) est une contrainte qui va au-delà du champ de lumière : il considère un cinquième degré de liberté en réfutant l'hypothèse (2) (la couleur d'un rayon est constant de tout son long). [Buehler *et al.*, 2001] ont intégré le critère 6 dans leur liste car il permet de corriger les erreurs de parallaxe et de projection des pixels directionnels survenant dans le cadre du champ de lumière plénoptique.

1.2 Nos contributions

Tout champ de lumière surfacique ne respecte pas les propriétés 2 (acquisition non structurée) et 8 (visualisation temps réel). Nos contributions permettent de respecter pleinement la propriété 2 et de gagner en temps de rendu.

1–Reconstruction robuste du champ de lumière surfacique. Dans [Wood *et al.*, 2000], une structure rigide est utilisée pour pouvoir répartir régulièrement les points de vue acquis. Ceci permet d’obtenir un échantillonnage local régulier et dense mais localement non complet du fait des auto-occultations. Dans [Coombe et Lastra, 2006], une résolution numérique complexe permet d’ajuster des polynômes à support compacts sur des échantillonnages incomplets. Ce procédé est uniquement guidé par la qualité visuelle et nécessite également des échantillonnages denses.

Notre contribution de reconstruction robuste permet de s’abstraire de toute contrainte de quantité et de répartition des acquisitions. Dans un premier temps, nous avons mis en avant les problèmes qu’implique un échantillonnage non structuré et non contrôlé. Nous avons alors proposé de régulariser l’ajustement de fonction en minimisant une énergie de stabilisation. La régularisation permet de définir un compromis entre précision et stabilité. Cela permet de générer un résultat déterministe, peu sensible aux aléas d’un échantillonnage non structuré et non contrôlé. Trois énergies de stabilisation ont été comparées : elles résolvent toutes le problème de sensibilité aux aléas mais seule une parmi elles, l’énergie de *Dirichlet*, génère un résultat souhaité indépendamment des choix qui ont pu être faits pour représenter la radiance.

Dans un second temps, nous avons évalué nos résultats numériquement grâce à la mise en place d’une mesure de l’erreur de prédiction. Cette mesure statistique permet d’évaluer la capacité d’une fonction ajustée à bien représenter un modèle de radiance dont on ne connaît qu’un (petit) échantillonnage. Grâce à cette mesure, un compromis optimal peut être calculé afin de régulariser suffisamment pour stabiliser l’ajustement mais suffisamment peu pour ne pas trop diminuer la précision.

En somme, cette contribution fournit un algorithme générant un résultat sans artefacts et plausible en tout état de cause. Si l’échantillonnage est bon (bien réparti et dense) et que la base est grande, le résultat sera de qualité élevée. Au contraire, si l’échantillonnage est de faible qualité à cause d’occultations, d’un manque de photos ou de bruit, le résultat sera « plausible » (*i.e.* il correspondra à une solution acceptable) et sans artefacts.

Ainsi, la structuration des acquisitions et leur quantité peut être quelconque : un résultat acceptable sera généré. À notre connaissance, de telles conditions n’ont jamais été considérées dans des travaux de reconstruction de champs de lumière. Ceci forme une première réponse complète au point 2 : ni structuration ni quantité minimale sont requises.

2–Simplification de maillages surfaciques avec attribut de radiance. La propriété 8 (rendu temps réel) est également un problème difficile auquel tout champ de lumière surfacique ne répond pas forcément. Tous les systèmes de rendu de champs de lumière surfaciques ont une limite de taille et de complexité des fonctions de radiance au-delà duquel il n’est pas possible de rendre en temps réel.

Nous avons proposé un outil permettant de déterminer un compromis entre qualité visuelle et taille des données. Cet outil se base sur la simplification par contractions d'arêtes. Nous y avons intégré une métrique sur la radiance d'un maillage. Cette métrique considère non seulement les fonctions de radiance stockées mais la radiance en tout point de la surface telle qu'elle sera rendue. Elle mesure ainsi des écarts entre radiances en un point de la surface avant et après la contraction d'une arête.

Les résultats montrent que cette métrique permet de préserver des caractéristiques que les techniques conventionnelles ne considèrent pas : les reflets directionnels, en particulier les pics spéculaires. Un champ de lumière surfacique peut ainsi être simplifié jusqu'à atteindre la taille ou la qualité souhaitée.

Cette contribution permet de repousser le seuil de rendu temps réel par le biais d'un choix de compromis qualité-complexité. Souvent un tel compromis est par ailleurs satisfaisant, notamment dans le cas d'écrans de taille réduite ou d'objets secondaires placés au fond d'une scène. En somme, cette contribution permet le respect de la propriété 8 si l'on simplifie suffisamment l'objet.

3–Visualisation de la radiance. La simplification des champs de lumière surfaciques engendre une surface moins dense où de grands triangles apparaissent. Un rendu naïf des fonctions de radiance génère alors une visualisation perfectible.

Notre contribution augmente le réalisme du rendu en assurant une bonne interpolation des fonctions de radiance. Elle définit la radiance en tout point du maillage par interpolation des fonctions, et non des vecteurs de coefficients de ces derniers. Pour cela, elle exploite la symétrisation de fonctions de radiance afin de pouvoir comparer et interpoler plus justement les reflets directionnels.

Nos résultats montrent que cette interpolation permet de rendre les reflets directionnels en tout point et d'obtenir une illusion de continuité d'ordre supérieur même avec un petit nombre de triangles. Remarquons que cette interpolation peut ralentir le rendu et qu'elle peut donc être contre-indiquée en cas d'impératifs de performance.

Bilan. En somme, le champ de lumière surfacique respecte de fait les propriétés 1, 3, 4, 5 et 7. Grâce à notre contribution de robustesse de la reconstruction, la propriété 2 est également respectée : il n'y a plus la moindre contrainte sur la structuration ni le nombre d'acquisitions photographiques, ce qui est une liberté bien plus forte que celle considérée dans [Buehler *et al.*, 2001]. Le respect de la propriété 8 dépend de la taille des données. Grâce à la simplification de modèles de radiance, elle peut désormais être respectée si l'on simplifie le modèle suffisamment. Finalement, la qualité du rendu d'un maillage non densément maillé peut être augmenté grâce à l'interpolation adéquate de radiance.

Toutes ces contributions peuvent fonctionner quelque soit le modèle choisi pour représenter la radiance : base de fonctions exprimés dans un repère local ou global, hémisphérique ou sphérique et couleurs définies dans un espace quelconque. Notons qu'au vu de l'utilisation de la représentation symétrisée dans les traitements survenant après la reconstruction, il n'est pas nécessaire de stocker la représentation non symétrisée des fonctions.

2 Choix du modèle de radiance

Tout au long de nos travaux, nous nous sommes autant que possible imposés de fournir des solutions fonctionnant pour tout modèle de radiance. Nous estimons en effet que l'utilisateur doit pouvoir librement faire ses choix en fonction de critères qui lui sont propres. En général, il s'agit des suivants :

- souhait de réalisme,
- besoin de compacité,
- capacité de calcul,
- bibliothèques compatibles, disponibles et/ou préférées,
- etc.

Nous avons néanmoins évoqué que certains choix facilitent certaines opérations. Nous en faisons ici une synthèse. L'utilisateur pourra ainsi intégrer ces éléments dans ses critères.

Représentation locale ou globale. Exprimer les fonctions dans un repère global (*e.g.* sphérique) plutôt que local a des avantages :

Visualisation. L'interpolation de fonctions peut se faire analytiquement à condition qu'elles soient toutes exprimées dans un repère unique donc global. Dans ce cas, interpoler les vecteurs de coefficients de fonctions a du sens.

Ceci implique cependant de construire des fonctions dans un domaine plus large que l'hémisphère de définition de la radiance, qui est orienté localement. L'interpolation s'appuie notamment sur la valeur des fonctions en-dehors de cet hémisphère et il faut donc s'assurer que l'hémisphère opposé ne contienne pas de valeurs erronées. L'ajustement stabilisé permet sans doute d'éviter de telles apparitions, mais une étude plus poussée mériterait d'être faite.

Simplification. Dans le cas d'un repère unique et global, l'erreur de contraction d'arêtes peut être écrite analytiquement et sous forme quadratique en les coefficients de la radiance qui sera attachée au sommet contracté. Ceci peut être fastidieux, mais permettrait d'obtenir les coefficients (*i.e.* la fonction de radiance) qui minimisent l'erreur en résolvant une équation linéaire.

En marge des étapes que nous avons traitées, une représentation globale est également plus facilement compressible. En effet, lorsqu'une fonction est symétrisée et définie dans un repère global, la distance entre fonctions peut être approximée par

la distance entre vecteurs de coefficients. Le problème de compression par *clustering* ou quantification devient alors un problème vectoriel et non plus fonctionnel.

Domaine ouvert ou fermé. Un domaine de paramétrage sphérique (ou homéomorphe à une sphère) n'a pas de bord (*i.e.* est fermé). Un tel domaine a l'avantage suivant :

Ajustement robuste. Les problèmes de robustesse sont réduits sur un domaine fermé : il n'y a pas de risque de générer des fonctions dont les valeurs deviennent extrêmes proche du bord du domaine. Dans le chapitre 4, nous avons vu que l'énergie E_2 était sujette à ces problèmes. Il est donc souhaitable d'exprimer les fonctions dans un domaine fermé si on veut l'utiliser.

Par sommet ou par texel. Le stockage de la radiance dans une texture offre de belles perspectives :

Simplification. L'avantage d'une représentation par texel est de pouvoir séparer la résolution du maillage et celle de la radiance (stockée dans une texture). L'outil proposé ne fonctionne que pour le stockage par sommet. Dans le cas de texels, un autre outil pourra être mis en place afin d'exploiter la séparation des résolutions (cf. perspectives du chapitre 6).

Visualisation. L'interpolation proposée au chapitre 5 est directement ré-utilisable pour interpoler des texels contenant des fonctions de radiance. Il faut cependant pouvoir interpoler analytiquement afin de pré-calculer une *mip-map*. Ceci n'est que possible si les fonctions sont exprimées dans un repère global.

Notons finalement qu'une texture nécessite un paramétrage et que l'interpolation à travers les bords des « *charts* » de la texture doivent être gérés.

Choix de la base. Le choix de la base (et de sa taille) n'influe pas sur la facilité de résolution des problèmes évoqués. Il régit cependant le besoin de simplification : plus la base est dense, plus les coûts mémoire et d'évaluation de fonctions de radiance sont élevés.

L'enjeu d'un tel choix porte surtout sur la capacité de représentation. D'expérience, nous choisissons parmi les fonctions (hémi-)sphériques (*e.g.* ondelettes sphériques, harmoniques sphériques, harmoniques hémisphériques, etc.) pour la représentation d'un champ de lumière surfacique. Elles ont une grande capacité de représentation et gèrent particulièrement bien les effets directionnels de la radiance. À nombre de coefficients égaux, leur qualité s'avère supérieure aux autres bases (*e.g.* polynômes et *Lumispheres*).

Bilan. La représentation en repère global facilite les calculs de comparaison et de distance sur les fonctions, opérations primordiales pour notre contexte. Dans ce cas

de repère global, il est possible d'utiliser un stockage par texel, ce qui pourrait être profitable dans une représentation à niveaux de détails.

En conclusion, si nous étions à la place d'un utilisateur souhaitant implémenter nos travaux en combinant une bonne capacité de représentation et posant le moins de problèmes possibles, nous choisirions la configuration suivante : harmoniques ou ondelettes sphériques en repère global, stockées dans une texture.

Ce choix est purement pragmatique et contextuel. En règle générale, c'est l'application visée qui déterminera le choix de la représentation.

Ainsi, les repères locaux sont souvent indiqués et utilisés car la génération de données sur un domaine sphérique est en réalité un non-sens. Il n'est en effet pas évident de déterminer les valeurs de la partie cachée, et cela peut avoir des implications selon les traitements envisagés, par exemple pour l'animation de maillages, où les orientations locales des hémisphères visibles sont modifiées.

Dans le cas d'objets translucides ou de fonctions de phase (le pendant volumique de la BRDF), ceci reprendrait du sens.

3 Perspectives

Nos travaux ont porté sur l'amélioration de la robustesse de reconstruction et sur la qualité et facilité de visualisation de champs de lumière surfaciques. Ceci contribue globalement à rendre la numérisation plus accessible et plus simple. Nous ne sommes cependant pas encore arrivés à une solution pleinement automatisée : le réglage du niveau de détail se fait par exemple encore manuellement et empiriquement. Les travaux que nous avons exposés ici permettent d'envisager plusieurs perspectives en continuité directe et permettant l'amélioration de la qualité et l'automatisation de l'utilisation des outils présentés.

3.1 Résumé des perspectives évoquées

Plusieurs perspectives ont déjà été mentionnées dans les chapitres respectifs. En voici une synthèse :

Reconstruction robuste. Nous ne traitons pas le cas des échantillonnages vides.

Ainsi, plusieurs points sont dépourvus de fonction de radiance. Une stratégie de diffusion spatiale de pixels directionnels permettrait de tirer profit de la cohérence entre voisins.

La diffusion spatiale pour remplissage de données manquantes a été bien étudié pour des images (donc de simples couleurs) [Bertalmio *et al.*, 2000; Barcelos et Batista, 2007]. Pour pouvoir transposer ces outils, les problèmes principaux sont le passage aux données fonctionnelles et surfaciques. Les problèmes sont donc de même nature que ceux que nous avons rencontrés dans ce manuscrit : passage du scalaire ou vectoriel au fonctionnel et gestion de repères

locaux. Les premières expérimentations faites dans le cadre d'un stage au sein de l'équipe IGG montrent que l'adaptation des outils « image » existants est compliquée.

Simplification. Nous avons argumenté qu'une évaluation numérique corrélée à la perception humaine mériterait d'être mise au point. Nous projetons donc de mettre en place une métrique globale d'écart de qualité entre un maillage avec radiance dégradé et une référence.

La nouveauté de cette métrique sera de mesurer un écart entre modèles visualisés, c'est-à-dire incluant les effets directionnels induits par la lumière et encodés dans des fonctions de radiance. Ceci n'a à notre connaissance pas encore été traité dans la littérature où seule de la géométrie est considérée, éventuellement avec des normales ou des couleurs. Nous exploiterons des écarts de radiance pour mettre en place cette métrique.

Elle sera finalement évaluée par des tests subjectifs permettant d'établir une corrélation avec la perception humaine. Pour cela, nous prévoyons d'établir un protocole d'évaluation basé sur des vidéos illustrant des maillages sous différents points de vue.

Visualisation et simplification. Nous n'avons pas mis en place de mécanisme d'interpolation ni défini de simplification sur une représentation de la radiance sous forme de texture. Il serait donc utile de compléter les travaux en testant l'interpolation de fonctions sur ces données avec des textures sous-échantillonnées notamment. Ensuite, nous pourrions définir un mécanisme de simplification exploitant la décorrélation entre la résolution géométrique et celle de la radiance.

3.2 Étude sur le choix des fonctions de radiance

Il n'existe à ce jour pas de consensus sur ce qui est la « meilleure » représentation fonctionnelle pour représenter de la radiance ou de la réflectance mesurée. De nouvelles représentations voient régulièrement le jour mais un état de l'art à ce sujet est manquant. En section 2, nous avons présenté un choix empirique possible, mais les critères sont nombreux.

Une étude numérique et perceptuelle serait utile afin d'évaluer précisément quels types de phénomènes réflectifs chaque base est capable de représenter. Il s'agirait d'aboutir à une mesure quantifiant le ratio *capacité de représentation sur mémoire utilisée* (*i.e.* nombre de coefficients).

Ceci n'est pas un problème facile car il faut pouvoir mettre en place une base de tests correspondant à de nombreux types de comportements réflectifs différents issus de numérisations réelles et simulées. Il est de plus difficile d'isoler les critères. Il faudrait par exemple pouvoir déterminer le coût mémoire d'une fonction à qualité égale et inversement.

Notons qu'un début d'étude a été faite pour les fonctions représentant des BTF [Filip et Haindl, 2009]. Les critères de qualité se ramènent à des mesures d'écart à une vérité terrain. Il serait intéressant de vérifier la corrélation de ces critères par rapport à la perception humaine.

3.3 Radiance par face

Nous souhaitons ouvrir une discussion en proposant d'attacher les données de colorimétrie (couleur, radiance, etc.) aux faces plutôt qu'aux sommets. Nous voyons deux arguments en faveur de ce choix.

Premièrement, cela paraît plus « naturel ». En effet, que ce soit l'étape d'acquisition (projection d'une portion de surface sur des pixels du capteur) ou de rendu (définition d'une couleur pour chaque portion de surface que couvre un pixel écran), l'entité en question est une portion de surface et non un point infinitésimal. Le contraire est vrai pour les mesures télémétriques mesurant la géométrie que nous avons utilisés. Ainsi, nous faisons à l'heure actuelle une double approximation : celle de la rétro-projection en un ensemble de points pré-définis et celle de l'interpolation de données définies qu'en un ensemble de points finis afin d'obtenir un rendu complet. La première est inévitable quelle que soit la structure utilisée (sommets, faces, texels). La seconde peut cependant être évitée en plaçant les données sur les faces : si toutes les faces sont colorées, alors toute la surface l'est.

Deuxièmement, cela permet de définir des *coupures franches*. L'indispensable interpolation des couleurs par sommet génère un flou lors du passage de coupures franches : le passage d'une rangée de sommets étiquetés d'une couleur à une rangée voisine de sommets étiquetés d'une autre nécessite au moins une rangée de triangles intermédiaires de transition où les couleurs sont interpolées (voir figure 7.1). Une solution est de sur-échantillonner la géométrie afin de réduire cette zone de transition. La définition par face évite cet écueil : une face est définie d'une couleur et sa voisine de l'autre, l'arête les reliant formant une frontière d'épaisseur nulle et donc nette. Ceci génère cependant une non continuité d'ordre 1 que nous abordons ci-dessous.

Verrous techniques. Grâce à la bibliothèque *CGoGN* [Kraemer *et al.*, 2013] nous disposons d'un outil privilégié et adéquat pour attacher des données quelconques aux faces : elle permet d'attacher tout type d'attribut à toute orbite de toute dimension (sommets, arêtes, faces, volumes). L'implémentation ne pose donc aucun souci.

Le passage dans le pipeline de numérisation reste cependant à définir.

Numérisation et reconstruction. Pour gérer cette étape, il faut rétro-projeter les pixels directionnels sur leur face d'origine au lieu de leur sommet. Chaque fonction est ensuite créée individuellement sur chaque face.

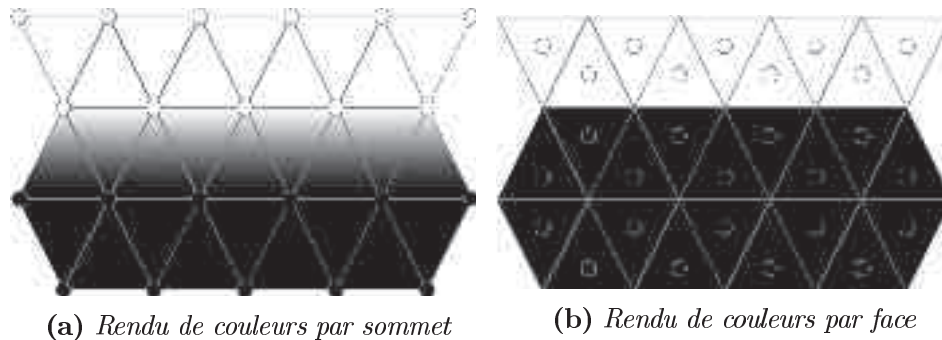


Figure 7.1 : La transition franche de couleurs n'est pas possible dans le cadre d'une interpolation de couleurs par sommet. Au moins une zone de triangles opère un dégradé pour passer d'une couleur à l'autre, ce qui peut donner un effet de flou. Le rendu de couleurs par face permet de déterminer des coupures franches.

Simplification. La simplification de maillage par contraction d'arête paraît applicable : deux faces, donc deux fonctions de radiance disparaissent. On souhaiterait alors mesurer l'impact de cette disparition et éventuellement ré-orienter la radiance des faces voisines dont la normale aura changée. Cet impact pourrait être composé d'un écart entre les fonctions de radiance qui disparaissent et celles de leurs faces adjacentes. Il n'y a pas de nouvel attribut de radiance à définir. La contraction géométrique avec placement optimal pourrait donc être envisagée.

Visualisation. Pour visualiser la surface, nous pourrions déterminer que chaque face ait pour couleur celle qui résulte de l'évaluation de sa fonction de radiance. Il n'y aurait ainsi pas d'interpolation à faire.

Cependant, si aucune interpolation n'est faite, l'illusion de surface lisse risque d'être perdue de façon analogue au rendu à plat. Inversement, on pourrait faire la supposition que la fonction de radiance est attachée au point central de la face puis interpoler avec ces voisines (par exemple sur le maillage dual). Ceci fait perdre le bénéfice de l'attache aux faces : il faut ré-instaurer un calcul d'interpolation et les coupures franches ne sont plus possibles.

Nous avons déjà établi un mode de rendu hybride tirant profit des deux modes de rendu, mais qu'appliqués aux couleurs diffuses. Ici à nouveau, il s'agit désormais de passer aux fonctions et de gérer les repères locaux.

Ainsi nous avons des premières pistes prometteuses pour aboutir à une solution utilisable de bout en bout. Les cartes graphiques actuelles ne traitent cependant que des propriétés par sommet. Il faut donc détourner quelque peu son utilisation en ayant recours à de la duplication de données par exemple (*i.e.* envoyer un sommet de multiples fois à la carte graphique, à savoir une fois par triangle incident et avec une couleur différente). Ceci implique un sur-coût mémoire qu'il conviendra d'estimer. Même si l'architecture des cartes graphiques n'est à ce jour pas faite pour ce nouveau modèle, nous estimons qu'il mérité d'être étudié afin de déterminer l'efficacité, la flexibilité et le réalisme qu'il est possible d'obtenir.

3.4 Reconstruction conjointe

En reconstruisant la géométrie et la photométrie de façon conjointe, nous pourrions générer un maillage adapté aux caractéristiques géométriques et visuels de l'objet. Après l'acquisition de l'un par radiométrie et de l'autre par prises de photos, on pourrait générer un maillage dont la densité est adaptée aux caractéristiques géométriques, mais aussi aux caractéristiques de radiance : un maillage sera ainsi plus dense, proche des fortes courbures géométriques ainsi que des fortes variations de radiance (couleur diffuse et réflexions directionnelles).

En règle générale, les appareils de mesure de la géométrie génèrent un nuage de points très dense. Cette géométrie est souvent simplifiée avant ou après la génération du maillage. Nous voudrions expérimenter la rétro-projection de pixels sur ce nuage de points afin d'y associer la radiance et d'ensuite en déduire un maillage adapté.

Une telle contribution permettrait de générer un maillage juste assez dense localement, et évitera d'avoir systématiquement recours à des outils de simplification. Actuellement, les premières passes de simplification (supprimant souvent plus de la moitié des sommets) ne génèrent aucun dégât visuel car le maillage est largement sur-échantillonné. Dans le cas d'une reconstruction adaptée, la simplification serait toujours utile mais uniquement lorsque l'on veut réellement dégrader la qualité au profit de la vitesse de la performance.

3.5 Long terme

Nous avons traité ou du moins acté plusieurs problèmes liés aux attributs fonctionnels d'un maillage, stockés par sommet ou par texel. On peut dès lors envisager de passer aux attributs de dimension supérieure : l'aspect sous forme bidirectionnelle. On peut par exemple penser aux fonctions de textures bidirectionnelles (*Bidirectional Texture Function* (BTF)) [Dana *et al.*, 1999]. Ce sont des textures numérisées dont chaque texel est une fonction bidirectionnelle : sa couleur est définie en fonction des directions hémisphériques d'une lumière incidente et d'une sortante (*i.e.* la direction d'observation). Il s'agit donc d'un espace 6D (deux coordonnées spatiales, et deux fois deux directionnelles). Elles sont acquises par de très nombreuses photos prises depuis des angles de vue différents et avec un angle d'éclairage variable également. La micro-géométrie et les effets de réflexions sous-surfaciques ne sont pas modélisés explicitement mais sont *de facto* inclus dans les acquisitions photographiques.

Les BTF sont très utilisés pour texturer des surfaces d'un matériau ayant des propriétés réfléchives réalistes. Le problème principal reste la quantité de données à traiter et à évaluer [Schwartz *et al.*, 2013], et la communauté cherche toujours à simplifier l'acquisition. Nous disposons des moyens pour acquérir de telles données [Ammann, 2010] et pourrions désormais nous intéresser à sa reconstruction et la simplification

ensuite, voire à la mise en place d'un mécanisme à niveaux de détails d'un maillage avec BTF. Pour cela, il reste à mettre en place une métrique adéquate, que l'on pourra espérer valider avec la métrique de qualité globale d'un maillage rendu.

QUATRIÈME PARTIE

ANNEXES

Algorithme de bootstrapping

Soit \mathcal{T} l'échantillonnage acquis d'une radiance donnée. La procédure de bootstrapping consiste à

- créer B bootstraps $\{T_1, \dots, T_B\}$. Chacun des T_b est un ensemble composé de N pixels directionnels choisis aléatoirement et avec remise (donc avec doublons) parmi ceux de T .
- Une fonction \hat{R}_b est ajustée sur chaque bootstrap T_b .
- Pour tout pixel directionnel $n \in T$, l'erreur quadratique est moyennée sur tous les bootstraps ne le contenant pas : c'est à dire pour tout R_b ajusté sur un ensemble privé de n . L'erreur est finalement composée de la moyenne de ce calcul sur tous les N pixels directionnels.

$$\hat{E} = \frac{1}{N} \sum_{n=1}^N \frac{1}{c_i} \sum_{b/n \notin T_b} \left(n^n - \hat{R}_b(\omega_r^n) \right)^2 \quad (\text{A.1})$$

avec $c_i = \#\{b/n \notin T_b\}$.

Remarquons qu'il n'est pas possible d'inverser les deux sommes, car le dénominateur c_i n'est connu qu'après tous les tirages.

Pour calculer \hat{E} grâce à cette procédure, l'algorithme (7) s'applique.

Algorithme 7 Algorithme de bootstrapping estimant l'erreur de prédiction

$$\hat{E} = \frac{1}{N} \sum_{n=1}^N \frac{1}{c_i} \sum_{b/n \notin T_b} \left(n^n - \hat{R}_b(\omega_r^n) \right)^2$$

```

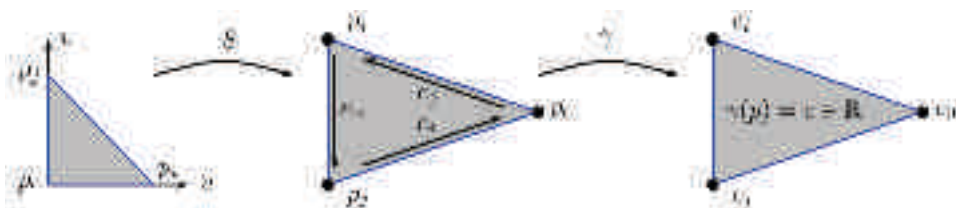
1:
2: function BOOTSTRAP( $\mathcal{T}, B$ )  ▷ Paramètre  $\mathcal{T}$  : l'échantillon complet (taille  $N$ )
3:                                     ▷ Paramètre  $B$  : le nombre de bootstraps
4:
Données:  $e$  qui est l'estimation  $\hat{E}$ 
Données:  $P$  l'échantillon de bootstrap (taille  $N$ )
Données:  $C$  le vecteur accumulant les  $c_i$  (taille  $N$ )
Données:  $E$  le vecteur accumulant les erreurs pour chaque échantillon (taille  $N$ )
Données:  $Oc$  le vecteur des occurrences de chaque échantillon dans  $P$  (taille  $N$ )
5:
6:    $C \leftarrow 0$ 
7:    $E \leftarrow 0$ 
8:   pour  $b$  de 1 à  $B$  faire                                     ▷ Pour chaque bootstrap
9:      $Oc \leftarrow 0$                                            ▷ Calculer l'échantillon
10:    pour  $i$  de 1 à  $n$  faire
11:       $k \leftarrow \text{Random}(1, n)$ 
12:       $P[i] \leftarrow \mathcal{T}[k]$ 
13:       $Oc[k] \leftarrow Oc[k] + 1$ 
14:    fin pour
15:     $\hat{R}_b \leftarrow \text{ajuster}(P)$                                      ▷ faire l'ajustement
16:    pour  $i$  de 1 à  $n$  faire                                     ▷ accumuler les erreurs
17:      si  $Oc[i] = 0$  alors
18:         $E[i] \leftarrow E[i] + \left( y_i - \hat{R}_b(x_i) \right)^2$ 
19:         $C[i] \leftarrow C[i] + 1$ 
20:      fin si
21:    fin pour
22:  fin pour
23:   $e \leftarrow 0$                                                ▷ calcul de l'erreur globale
24:  pour  $i$  de 1 à  $n$  faire
25:    si  $C[i] > 0$  alors
26:       $e \leftarrow e + E[i]/C[i]$ 
27:    fin si
28:  fin pour
29:   $e \leftarrow e/n$ 
30:  retourner  $e$ 
31: fin function
32:

```

Annexe **B**

Gradient sur un triangle

Nous allons définir le gradient d'une fonction scalaire sur le triangle dans le cas où cette fonction est le résultat d'une interpolation bi-linéaire de scalaires définis aux sommets du triangle. Considérons l'application $\gamma : p \mapsto c \in \mathbb{R}$, transformant un point quelconque du triangle en un scalaire. Le gradient n'étant pas calculable directement sur le triangle, on s'en donne un paramétrage $S(u, v)$. Le résultat sera au final indépendant de ce paramétrage.



On a alors :

$$S(u, v) = u \cdot p_i + v \cdot p_j + (1 - u - v) \cdot p_0$$
$$\gamma \circ S(u, v) = u \cdot c_i + v \cdot c_j + (1 - u - v) \cdot c_0$$

Le gradient de γ sur cette paramétrisation se définit alors par [Do Carmo, 1976] :

$$\nabla \gamma(S(u, v)) = J_S (J_S^T J_S)^{-1} \nabla (\gamma \circ S)(u, v) \quad (\text{B.1})$$

où

- $J_S = J(S(u, v))$ est la Jacobienne $[\frac{\partial S(u, v)}{\partial u} \frac{\partial S(u, v)}{\partial v}]_{3 \times 2} = [(p_i - p_0), (p_j - p_0)] = [\mathbf{e}_j, -\mathbf{e}_i]$.
- $\nabla(\gamma \circ S)(u, v) = \begin{bmatrix} \frac{\partial(\gamma \circ S)(u, v)}{\partial u} \\ \frac{\partial(\gamma \circ S)(u, v)}{\partial v} \end{bmatrix} = \begin{bmatrix} c_i - c_0 \\ c_j - c_0 \end{bmatrix}$

On obtient alors :

$$J_S^T J_S = \begin{bmatrix} \mathbf{e}_j^T \\ -\mathbf{e}_i^T \end{bmatrix} [\mathbf{e}_j \quad -\mathbf{e}_i] = \begin{bmatrix} \|\mathbf{e}_j\|^2 & \langle \mathbf{e}_j, -\mathbf{e}_i \rangle \\ \langle -\mathbf{e}_i, \mathbf{e}_j \rangle & \|\mathbf{e}_i\|^2 \end{bmatrix}_{2 \times 2} \quad (\text{B.2})$$

Le déterminant de la matrice (B.2) est non nul lorsque l'aire A_T du triangle est non nulle :

$$\text{Det}(J_S^T J_S) = \|\mathbf{e}_j\|^2 \|\mathbf{e}_i\|^2 - \langle \mathbf{e}_j, -\mathbf{e}_i \rangle \langle -\mathbf{e}_i, \mathbf{e}_j \rangle = \|\mathbf{e}_j \wedge \mathbf{e}_i\|^2 = (2A_T)^2 = 4A_T^2 \quad (\text{B.3})$$

Ainsi, une précondition au calcul du gradient est de considérer un triangle d'aire non nulle. Si cette condition est bien vérifiée, on obtient :

$$J_S^T J_S^{-1} = \begin{bmatrix} \|\mathbf{e}_i\|^2 & \langle \mathbf{e}_i, \mathbf{e}_j \rangle \\ \langle \mathbf{e}_i, \mathbf{e}_j \rangle & \|\mathbf{e}_j\|^2 \end{bmatrix}_{2 \times 2} \quad (\text{B.4})$$

et ainsi :

$$J_S(J_S^T J_S^{-1}) = \frac{1}{4A_T^2} [\|\mathbf{e}_i\|^2 \mathbf{e}_j - \langle \mathbf{e}_i, \mathbf{e}_j \rangle \mathbf{e}_i, \langle \mathbf{e}_i, \mathbf{e}_j \rangle \mathbf{e}_j - \|\mathbf{e}_j\|^2 \mathbf{e}_i]_{3 \times 2} \quad (\text{B.5})$$

On peut alors déduire que le gradient vaut

$$\nabla \gamma(S(u, v)) = \frac{1}{4A_T^2} [\|\mathbf{e}_i\|^2 \mathbf{e}_j - \langle \mathbf{e}_i, \mathbf{e}_j \rangle \mathbf{e}_i, \langle \mathbf{e}_i, \mathbf{e}_j \rangle \mathbf{e}_j - \|\mathbf{e}_j\|^2 \mathbf{e}_i] \begin{bmatrix} c_i - c_0 \\ c_j - c_0 \end{bmatrix} \quad (\text{B.6})$$

ce qui est équivalent à

$$\frac{1}{4A_T^2} [(\|\mathbf{e}_i\|^2 |c_i - c_0| + \langle \mathbf{e}_i, \mathbf{e}_j \rangle (c_j - c_0)) \mathbf{e}_j - (\|\mathbf{e}_j\|^2 (c_j - c_0) + \langle \mathbf{e}_i, \mathbf{e}_j \rangle (c_i - c_0)) \mathbf{e}_i] \quad (\text{B.7})$$

Génération à la volée de textures multi-échelles par l'exemple

Cette annexe présente un travail auquel nous avons pu contribuer et qui a donné lieu à une présentation dans la conférence *Siggraph Asia 2013* ainsi qu'à une publication dans la revue *ACM Transactions on Graphics* [Vanhoey *et al.*, 2013b].

Comparé aux travaux de ce manuscrit, ce travail est de nature toute autre, c'est pourquoi nous ne le présentons qu'en annexe. Il est le fruit d'une collaboration entre les quatre auteurs : Jean-Michel DISCHLER est l'initiateur du projet et nous avons pu contribuer en :

- proposant un algorithme de fragmentation de texture,
- participant aux nombreuses discussions qui ont abouti aux deux contributions principales : les méthodes de calcul de contenu échangeable et de transfert de couleur entre les échelles de texture tels que présentées dans l'article,
- participant activement à la rédaction.

Nous résumons brièvement les contributions de l'article et présentons ensuite l'algorithme de fragmentation et sa justification. L'article peut être consulté pour une vision complète de ces travaux, comprenant notamment le positionnement par rapport à l'existant.

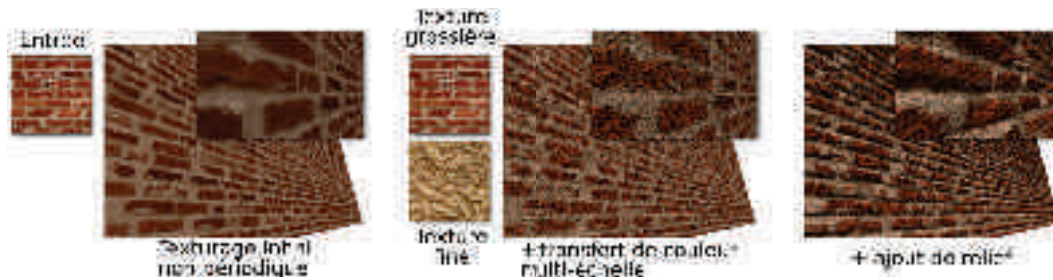


Figure C.1 : Résultats des trois contributions de l'article.

1 Introduction & contributions

Ces travaux se placent dans le domaine du rendu, plus particulièrement dans celui de la génération de textures par l'exemple. L'article comprend trois contributions illustrées dans la figure C.1. À partir d'une (petite) texture, elle offre les possibilités suivantes :

- générer un pavage infini par recopie de cette texture : à chaque instance recopiée, le contenu du nouveau pavé est modifié pour que ni la répétition ni la grille de pavage soit perceptible ;
- transiter naturellement entre plusieurs niveaux d'échelle de textures sans artefacts (ceci nécessite une texture par niveau en entrée) ;
- extraire du relief d'une texture de façon intuitive et rapide, puis utiliser ce relief pour augmenter le réalisme du rendu.

La modification du contenu et la transition entre niveaux d'échelle se fait à la volée sur la carte graphique, grâce à des informations pré-calculées.

Pour ce pré-calcul, deux outils sont exploités : un espace de couleur permettant de séparer un certain nombre de couleurs principales de la texture et des variations locales ainsi qu'une fragmentation de l'image. Nous avons plus particulièrement mis au point cette dernière étape que nous détaillons ci-dessous.

2 Les fragments

La fragmentation est un composant essentiel du système mis en place. Son but est d'isoler des parties de l'image comprenant une caractéristique particulière (*e.g.* une pétale de fleur dans la figure C.4) et dont le bord correspond à de hauts contrastes dans la texture.

Ces fragments définissent ainsi des éléments visuels saillants aux yeux de l'observateur : c'est avant tout la structure des fragments qui caractérise la texture. Ainsi, on voudra que dans tous les traitements que l'on appliquera (calcul de nouveau contenu pour un pavé, modification de la couleur pour la mise en cohérence des textures de



(a) *Modification de couleur aléatoire.*



(b) *Modification de couleur par fragment.*

Figure C.2 : *Comparaison entre modification de contenu quelconque ou par fragment.*

différentes échelles, application de *displacement* ou *relief mapping*), un fragment ne soit jamais coupé voire doit être préservé à tout prix.

De plus, la propriété d'avoir un bord à haut contraste permet de bénéficier de l'effet de *texture masking* : modifier le contenu près de ces bords est moins visible car le bord à haut contraste est en soi un élément saillant. Ceci est illustré par la figure C.2 où le changement de couleur suivant des bords d'éléments à forts contrastes paraît plus naturel. L'échange de contenu et le mélange multi-échelle exploitent cette propriété.

2.1 Mélange multi-échelle

Lors d'un rendu multi-échelle avec deux textures, un texel d'une texture grossière se superpose à une empreinte de $n \times n$ texels d'une texture fine, où n est le facteur d'échelle choisi. Pour éviter l'apparition d'artefacts de couleur, il est nécessaire de les modifier afin que la moyenne des $n \times n$ texels du niveau fin soit égale à la couleur du texel du niveau grossier. Une façon de faire est d'opérer une translation dans l'espace des couleurs sur les texels du niveau fin. Ceci génère l'effet de la figure C.3a. Souvent, un filtrage est appliqué afin de diminuer l'effet « en blocs », mais la sensation que deux couleurs ou motifs distincts ont été superposés est encore présent (figure C.3b).

Nous proposons une autre façon d'établir cette cohérence de couleurs, en exploitant les fragments. Ceci génère le résultat de la figure C.3c où l'on donne l'impression qu'une couleur et un motif grossier sont constitués de couleurs et motifs à l'échelle plus fine.

Pour ce faire, nous modifions la translation dans l'espace de couleurs, de façon à ce que non pas le carré de $n \times n$ texels ait pour couleur moyenne la couleur du texel grossier correspondant, mais que l'ensemble des texels d'un fragment ait pour moyenne celle du texel grossier correspondant à *son texel central*. Les couleurs sont ainsi « transférés » à travers les échelles *par fragment*.

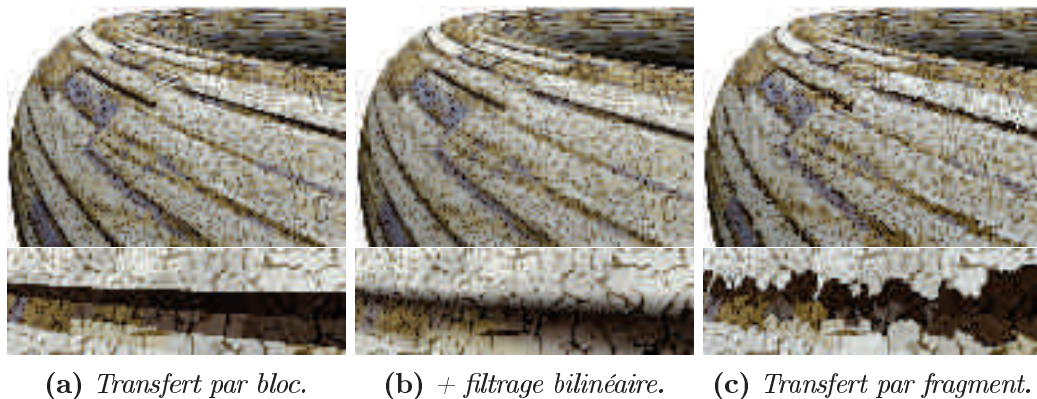


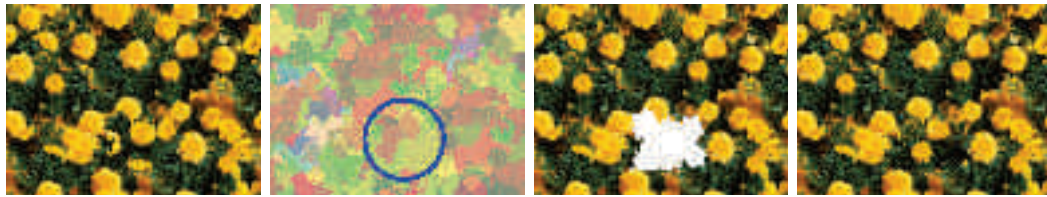
Figure C.3 : Mélange multi-échelle de couleurs. Le transfert par blocs (la couleur d'un texel grossier est égal à la moyenne des couleurs des $n \times n$ texels fins qu'il recouvre) génère des artefacts (figure C.3a). Un filtrage permet de cacher l'effet de blocs (figure C.3b) mais ne donne pas le réalisme escompté. Le transfert par fragment (la couleur d'un texel grossier est égal à la moyenne des couleurs des texels dont le fragment a pour centre un texel recouvert) génère l'effet souhaité où la texture fine semble constituer la texture grossière.

Notons que ce sont les propriétés des fragments (englober une seule caractéristique et avoir un bord à haut contraste) qui font en sorte que le transfert de couleur suive les bords des structures de la texture fine et ne génère ainsi pas ou peu d'artefacts.

2.2 Contenu inter-changeable

Pour générer une texture « infinie », la même texture périodique fournie en entrée¹ est répétée en un pavage régulier. Pour éviter de détecter une répétition dans le pavage généré lors du rendu, la technique mise en place consiste à échanger le contenu de certaines zones pré-déterminées de la texture (en général une dizaine), appelés *patches*, et dont on aura pré-calculé différents contenus. Est considéré comme potentiel contenu pour un patch, toute zone de même forme prise à un autre endroit dans la même texture. Il y a autant de candidats que de texels et nous en gardons quelques-uns (en général une demi douzaine) : ceux qui minimisent une métrique d'erreur visuelle. Il suffit lors du rendu de choisir aléatoirement, pour chaque patch, un des contenus pré-déterminés afin de remplacer le contenu original. La figure C.4 illustre ce procédé. Ainsi, les textures générées lors du pavage vont avoir un contenu aléatoire et on peut alors générer un nombre de pavés différents égal au nombre de patches puissance le nombre de contenus pré-calculés.

1. Une par échelle.



(a) Image d'origine. (b) Fragmentation. (c) Patch. (d) Contenu échangé.

Figure C.4 : Échange de contenu. À partir de la fragmentation de l'image d'entrée (figure C.4a), des formes de patches sont définies par « circle packing » (figure C.4b). Plusieurs nouveaux contenus sont pré-calculés pour chaque patch. Lors du rendu d'un pavé, un de ces contenus est choisi aléatoirement pour venir remplir le patch (figure C.4d).

Forme des patches. Un patch doit avoir un bord à haut contraste afin de masquer l'échange de contenu (*i.e.* les erreurs aux bords). Or, contenir qu'une seule caractéristique ne suffit pas : échanger deux pétales de fleurs dans la figure C.4a ne permet pas de briser la répétition. Il est plus judicieux qu'un patch rassemble plusieurs caractéristiques (*e.g.* quelques fleurs entourés d'un peu d'herbe). Ainsi, l'échange de contenu permettra de changer la structure complète de l'image en un patch : quelques fleurs seront remplacées par des fleurs d'autres formes ou couleurs par exemple.

Au vu de ces besoins, nous définissons la forme des patches par « *circle packing* » : on se donne un cercle que l'on place en divers endroits de la texture et on agglomère les fragments qui se trouvent en son sein (voir figure C.4b). Ainsi, on obtient bien des patches composés de plusieurs caractéristiques et avec des hauts contrastes aux bords.

Contenu des patches. Afin de déterminer une liste de contenus pouvant remplacer le contenu initial d'un patch, on a recours à une métrique d'erreur, que nous avons défini empiriquement. Il s'agit de mesurer la discontinuité engendrée au bord d'un patch par un échange de contenu. Nous nous basons sur l'hypothèse que des séquences de fortes discontinuités consécutives sont saillantes. Ainsi, nous avons défini une métrique I exploitant un filtre de convolution amplifiant les gradients le long du



Figure C.5 : Deux contenus différents pour un même patch de forme circulaire.

bord.

La figure C.5 illustre sa pertinence : le contenu de gauche est la solution minimisant l'erreur moyenne des écarts quadratiques le long du bord. Or, cette solution montre de faibles discontinuités le long de la moitié inférieure du bord et de fortes discontinuités consécutives le long de la moitié supérieure. À droite se trouve la solution minimisant notre métrique pénalisant les erreurs consécutives (et qui a une erreur quadratique moyenne supérieure à la solution de gauche). Il est évident que la solution de droite est préférable.

En somme, en choisissant une forme de patch qui épouse les forts contrastes, puis en choisissant les contenus minimisant I , l'échange de contenu est quasiment imperceptible. De plus, la carte des patches peut être stockée efficacement sur la carte graphique : une texture supplémentaire stocke l'identifiant d'un patch (*i.e.* sa forme) et un tableau global stocke, pour chaque patch, la liste des nouveaux contenus pré-calculées, parmi lesquels il conviendra de choisir aléatoirement. Cette métrique étant empirique, elle peut sans doute être améliorée. De réels critères perceptuels permettront d'intégrer plus amplement les notions de saillance visuelle.

2.3 Algorithme de fragmentation

La fragmentation détermine les zones sensées correspondre à des caractéristiques visuelles et dont les bords coïncident avec de forts contrastes. Elle prends en entrée la transformée de l'image en un nouvel espace de couleurs. Chaque texel y est exprimée comme une couleur dominante D_i et une coordonnée v le long d'un vecteur de variation V_i . Il n'y a que quelques couples (D_i, V_i) par image, stockée globalement. Ainsi, la couleur d'un texel est déterminée par le couple (i, v) . On peut y penser comme une couleur issue d'une factorisation et une variation locale.

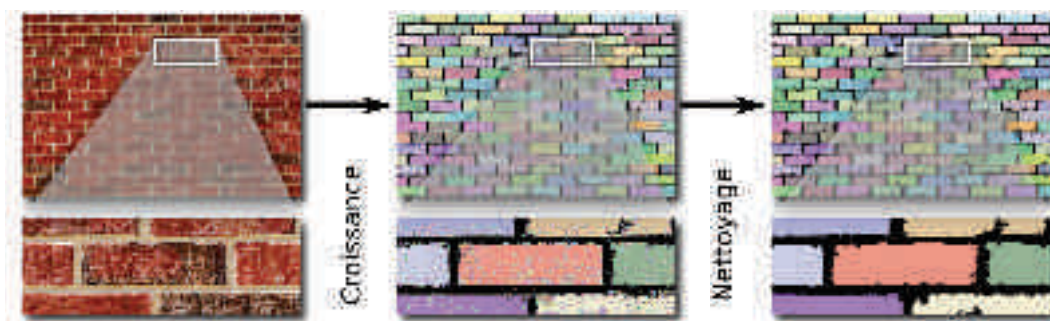


Figure C.6 : *Fragmentation : l'image est découpée en plusieurs fragments dont le contenu est homogène et dont les bords sont à forts contrastes.*

L'algorithme (8) procède alors en deux étapes (voir figure C.6). Dans un premier temps, un algorithme de croissance de régions aboutit à des fragments couvrant

complètement la texture. Dans un second temps, les plus petits fragments sont « nettoyés ».

Algorithme 8 Algorithme de fragmentation.

Une texture T est donnée en entrée, ainsi que trois paramètres :

- un seuil de contraste τ ,
- une taille de fragment minimale F_{min} ,
- une taille de fragment maximale F_{max} .

Croissance de régions. Des régions F_k grandissent autour d'une graine aléatoire.

Un texel \mathbf{p} est ajouté à F_k si et seulement si :

1. c'est un 4-voisin de F_k ;
2. son indice de couleur $i(\mathbf{p})$ est identique à celui de la région en croissance ;
3. sa coordonnée variationnelle ne dépasse pas le seuil de contraste par rapport à la moyenne des texels de la région courante : $\|v(\mathbf{p}) - \frac{1}{|F_k|} \sum_{\mathbf{q} \in F_k} v(\mathbf{q})\| < \tau$.

Lorsqu'aucun texel ne respecte ces contraintes, ou lorsque F_{max} texels composent la région courante, celle-ci est fermée et définit désormais un fragment. L'algorithme itère alors en répétant ce processus en prenant pour graine un texel aléatoire parmi ceux qui sont incidentes aux bords des fragments existants, jusqu'à ce que tous les texels soient pris en compte.

Nettoyage de fragments. Certains fragments peuvent être plus petits que F_{min} , notamment à cause de variations hautes fréquences dans l'image. C'est pourquoi nous introduisons un nettoyage qui vide un à un les fragments trop petits en ordre de taille croissante. Cette opération consiste à dispatcher les texels dans les fragments voisins, plus particulièrement dans celui qui est le plus proche en termes de coordonnée variationnelle moyenne.

2.4 Propriétés importantes

La fragmentation résultante a les propriétés suivantes. Premièrement, tous les fragments ont des tailles proches. Deuxièmement, tous les texels d'un fragment ont la même couleur dominante et la même variation principale. Troisièmement, les coordonnées variationnelles d'un fragment sont relativement uniformes. Tout ceci implique que les fragments ont tendance à emprisonner des caractéristiques visuelles et à avoir de forts contrastes en leurs bords, permettant d'exploiter le « texture masking » dans le transfert de couleur et l'échange de contenu.

De plus, le stockage de la carte de fragments est pratique en vue d'un rendu temps réel : dans une texture complémentaire, nous stockons en tout texel le vecteur de translation du texel courant vers le centre de son fragment d'appartenance. Ainsi, lors du rendu, retrouver le fragment de tout texel est réalisé par une indirection.

3 Conclusion

La contribution principale de ce travail est d'établir un juste équilibre entre pré-calculs et compacité des données à transmettre à la carte graphique. Grâce au principe d'échange de contenu, réfléchi pour cacher d'éventuels artefacts, un grand nombre de pavés différents peut être généré. Grâce à une mise en cohérence profitant des mêmes caractéristiques (masquage d'artefacts), le transfert de couleur multi-échelle semble naturel. Ce sont les fragments et leurs caractéristiques qui permettent d'obtenir ces résultats. Finalement, c'est grâce à la compacité de la représentation des fragments et des patches que le rendu peut se faire à faible coût mémoire et calculatoire, donc en temps réel.

3.1 Perspectives en lien avec nos travaux de thèse

Ces travaux étant sans lien apparent direct avec ceux décrits dans ce manuscrit, l'article [Vanhoey *et al.*, 2013b] ne mentionne pas de perspectives liées. Nous pouvons cependant projeter quelques extensions en relation avec cette thèse, par exemple le passage à des textures multi-dimensionnelles. On peut ainsi imaginer un passage aux textures polynomiales (PTM) ou aux BTF voire aux textures dynamiques (*i.e.* évoluant dans le temps).

Pour appliquer la contribution d'échange de contenu à la volée, il faut pouvoir pré-calculer la forme des patches et une liste de contenus. Pour ce faire, nous avons fait appel à une fragmentation pour définir la forme, et à une métrique d'erreur mesurant l'approximation commise aux bords des patches en cas d'échange de contenu. Il faut ainsi pouvoir adapter ces deux méthodes à des texels contenant une information fonctionnelle, et non scalaire.

Dans les deux cas (fragmentation et métrique d'erreur), il s'agit d'établir des métriques pertinentes, mesurant des écarts entre texels. Dans le cas des PTM, nous disposons de tous les outils nécessaires : notamment l'application d'une distance entre fonctions de radiance. Pour les BTF, le principe ne diffère pas grandement, mais quelques travaux additionnels seraient nécessaires en vue du passage au bi-directionnel. Finalement, pour les textures dynamiques, le problème est plus ardu : les endroits saillants évoluant au cours du temps, il est difficile de faire un découpage de patches statique. Une réflexion plus poussée devra être menée afin de déterminer les solutions adaptées.

Liste des figures

1.1	Numérisation : exemple d'application	6
1.2	Maillage surfacique dense	8
1.3	Exemple de champ de lumière	9
1.4	Schéma de reconstruction	10
1.5	Scanners 3D	12
1.6	Constellation locale de pixels directionnels	16
2.1	Fonction plénoptique	26
2.2	Mosaïques concentriques	30
2.3	Descripteurs de couleur ou de réflectance	33
2.4	Exemples de BRDF	38
3.1	Paramétrage du champ de lumière plénoptique	46
3.2	Exemples de champ de lumière plénoptique	47
3.3	Montages de caméras	48
3.4	Paramétrages du champ de lumière plénoptique	49
3.5	Demi-espace de visibilité d'un point	54
3.6	Illustration des espaces de couleur	57
3.7	Exemples de bases de fonctions de radiance	63
4.1	Diversité d'échantillonnage dans une même acquisition	74
4.2	Ajustement séparée de fonctions	76
4.3	Ajustement de fonctions : comparaison des normes	79
4.4	Défaut de robustesse	81
4.5	Ré-échantillonnage des directions d'observation	82
4.6	Propriétés des énergies	85
4.7	Jeu de test	87
4.8	Exemples de stabilisation	88
4.9	Comparaison des énergies : échelle mésoscopique	89
4.10	Comparaison des énergies : <i>Stanford bunny</i>	91
4.11	Comparaison des énergies : <i>double dragon</i>	91
4.12	Comparaison des énergies : espaces de couleur et base	93
4.13	Reconstruction du masque : N variable	95
4.14	Reconstruction du masque : K variable	95
4.15	Effet de stabilisation de l'ajustement	97
4.16	Graphe d'erreur de prédiction type	101
4.17	Graphes d'erreurs de prédiction selon diverses variantes	104
4.18	Graphes de robustesse au changement de base	105

4.19	Pourcentage de points nécessitant une forte stabilisation	106
4.20	Polynomial texture maps	109
4.21	Ré-éclairage : le « <i>Capitello</i> »	110
5.1	Structure de données	114
5.2	Pipeline GPU	117
5.3	Rendus simulés	119
5.4	Rendus de couleurs par sommet	122
5.5	Interpolation classique de la radiance	123
5.6	Radiance spéculaire	124
5.7	Symétrisation de la radiance	126
5.8	Rendu de radiance symétrisé : comparaisons sur données synthétiques	131
5.9	Rendu de radiance symétrisé : comparaisons sur données numérisées	132
6.1	Contraction d'arête	141
6.2	QEM : lignes de niveau et variante.	143
6.3	Simplification avec QEM	144
6.4	Semi-contraction	150
6.5	Illustration de E_{col} sur exemples synthétiques	151
6.6	Comparaison : semi-contraction <i>versus</i> placement optimal	153
6.7	Comparaison : simplifications sur critères de couleur (objet <i>Masque</i>)	154
6.8	Comparaison : simplifications couleur <i>vs</i> radiance (objet synthétique)	158
6.9	Comparaison : simplifications couleur <i>versus</i> radiance (objet <i>Masque</i>)	160
6.10	Simplifications successives du <i>double dragon</i>	161
6.11	Métrique d'erreur de radiance quadratique	164
7.1	Rendu par sommet ou par face	178
C.1	Texturage infini multi-échelle avec relief	188
C.2	Modification de contenu par fragment	189
C.3	Mélange multi-échelle de couleurs	190
C.4	Pré-calcul pour l'échange de contenu : la forme	191
C.5	Pré-calcul pour l'échange de contenu : le contenu	191
C.6	Fragmentation	192

Liste des tableaux

3.1	Taxinomie et comparaison des espaces de couleur	61
3.2	Comparaison des bases de fonctions	66
7.1	Propriétés du rendu SLF	170

Liste des algorithmes

1	<i>Shaders</i> pour le rendu à plat	119
2	<i>Shaders</i> pour le lissage de <i>Gouraud</i>	120
3	<i>Shaders</i> pour l'interpolation de <i>Phong</i>	121
4	<i>Geometry shader</i> préparant l'interpolation de radiance	128
5	<i>Fragment shader</i> d'interpolation de radiance	129
6	Algorithme de contraction d'arêtes	141
7	Algorithme de bootstrapping	184
8	Algorithme de fragmentation	193

Lexique

Ajustement	73
Algorithme d'ajustement robuste	82
Biais	77
Biais de stabilisation	98
Biais modèle	77
Capacité de représentation	62
Compromis	100
Compromis optimal	100
E_{MSE}	80
Échantillonnage creux	80
Échantillonnage dense	80
Échantillonnage très creux	80
Échantillonnage	73
Échelle macroscopique	80
Échelle mésoscopique	80
Ensemble d'entraînement	100
Ensemble de validation	100
Erreur de prédiction	99
Espace perceptuellement uniforme	56
Fonction interpolante	77
<i>Fragment shader</i>	116
<i>Geometry shader</i>	116

Geomorphing	140
Interpolation fonctionnelle	129
Interpolation scalaire	129
Lobe spéculaire	38
Luminance lumineuse sortante	9
Luminance énergétique sortante	9
Maillage progressif	148
Mip-map	134
Modèle de fonction	62
Photométrie	7
Pixels directionnels	17
Précision	75
Quadrique	143
Quantité photométrique	9
Quantité radiométrique	9
Radiance sortante	9
Raffinement adaptatif	148
Régression	100
Régularisation	100
Résultat imprévisible	75
Résultat inattendu	75
Résultat prévisible ou déterministe	75
Résultat souhaité ou attendu	75
Semi-contraction	147
<i>Shader</i>	116
Solution au sens des moindres carrés	80
Somme de quadriques	143
Stabilité	75
Symétrisation	125
<i>Vertex shader</i>	116

BIBLIOGRAPHIE

ADELSON, E. H. et BERGEN, J. R. (1991). The plenoptic function and the elements of early vision. *In Computational Models of Visual Processing*, pages 3–20. MIT Press.

AMMANN, L. (2010). *Visualisation temps réel de données à deux dimensions et demie*. Thèse de doctorat, Université de Strasbourg.

ARBACE, L., SONNINO, E., CALLIERI, M., DELLEPIANE, M., FABBRI, M., IDELSON, A. I. et SCOPIGNO, R. (2013). Innovative uses of 3d digital technologies to assist the restoration of a fragmented terracotta statue. *Journal of Cultural Heritage*, 14(4):332 – 345.

ASHIKMIN, M., PREMOŽE, S. et SHIRLEY, P. (2000). A microfacet-based brdf generator. *In Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 65–74, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

BAO, S. Y., CHANDRAKER, M., LIN, Y. et SAVARESE, S. (2013). Dense object reconstruction with semantic priors. *In Proceedings of the 2013 IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1264–1271, Los Alamitos, CA, USA. IEEE Computer Society.

BARCELOS, C. A. Z. et BATISTA, M. A. (2007). Image restoration using digital inpainting and noise removal. *Image and Vision Computing*, 25(1):61 – 69. SIBGRAPI.

- BECKMANN, P. et SPIZZICHINO, A. (1987). *The Scattering of Electromagnetic Waves from Rough Surfaces (Artech House Radar Library)*. Artech Print on Demand.
- BERNARDINI, F. et RUSHMEIER, H. (2002). The 3d model acquisition pipeline. *Computer Graphics Forum*, 21(2):149–172.
- BERTALMIO, M., SAPIRO, G., CASELLES, V. et BALLESTER, C. (2000). Image inpainting. *In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, pages 417–424, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- BESL, P. et MCKAY, N. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256.
- BLINN, J. F. (1977). Models of light reflection for computer synthesized pictures. *In Proceedings of the 4th annual conference on Computer graphics and interactive techniques, SIGGRAPH '77*, pages 192–198, New York, NY, USA. ACM.
- BOTSCH, M., KOBBELT, L., PAULY, M., ALLIEZ, P. et LEVY, B. (2010). *Polygon Mesh Processing*. AK Peters.
- BOTSCH, M., STEINBERG, S., BISCHOFF, S. et KOBBELT, L. (2002). Openmesh – a generic and efficient polygon mesh data structure. *In OpenSG Symposium*.
- BOUBEKEUR, T. et ALEXA, M. (2008). Phong tessellation. *ACM Transactions on Graphics*, 27(5):141:1–141:5.
- BUEHLER, C., BOSSE, M., McMILLAN, L., GORTLER, S. et COHEN, M. (2001). Unstructured lumigraph rendering. *In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH '01*, pages 425–432, New York, NY, USA. ACM.
- CALLIERI, M., SCOPIGNO, R. et SONNINO, E. (2011). Using 3d digital technologies in the restoration of the madonna of pietranico. *ERCIM News*, 2011(87).
- CAMAHORT, E., ABAD, F. et FUSSELL, D. (2009). A line-space analysis of light-field representations. *Graphical Models*, 71(5):169–183.
- CAMAHORT, E., LERIOS, A. et FUSSELL, D. (1998). Uniformly sampled light fields. *In Rendering Techniques '98*, Eurographics, pages 117–130. Springer Vienna.
- CGAL OPEN SOURCE PROJECT (1996). CGAL. <http://www.cgal.org/>.
- CHAI, J.-X., TONG, X., CHAN, S.-C. et SHUM, H.-Y. (2000). Plenoptic sampling. *In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*, pages 307–318, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

CHEN, S. E. (1995). Quicktime vr: an image-based approach to virtual environment navigation. *In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 29–38, New York, NY, USA. ACM.

CHEN, W.-C., BOUGUET, J.-Y., CHU, M. H. et GRZESZCZUK, R. (2002). Light field mapping: efficient representation and hardware rendering of surface light fields. *In Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 447–456, New York, NY, USA. ACM.

COHEN, J., OLANO, M. et MANOCHA, D. (1998). Appearance-preserving simplification. *In SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 115–122, New York, NY, USA. ACM.

COMMISSION INTERNATIONALE DE L'ÉCLAIRAGE (CIE) (1976). Cieluv and cielab color spaces.

COMMISSION INTERNATIONALE DE L'ÉCLAIRAGE (CIE) (1988). Spectral luminous efficiency function for photopic vision.

COOK, R. L. et TORRANCE, K. E. (1982). A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24.

COOMBE, G., HANTAK, C., LASTRA, A. et GRZESZCZUK, R. (2005). Online construction of surface light fields. *In Rendering Techniques*, pages 83–90. Eurographics Association.

COOMBE, G. et LASTRA, A. (2006). An incremental weighted least squares approach to surface lights fields. *In GRAPP*, pages 84–91. INSTICC - Institute for Systems and Technologies of Information, Control and Communication.

CORSINI, M., DELLEPIANE, M., PONCHIO, F. et SCOPIGNO, R. (2009). Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties. *Computer Graphics Forum*, 28(7):1755–1764.

CORSINI, M., LARABI, M. C., LAVOUÉ, G., PETŘÍK, O., VÁŠA, L. et WANG, K. (2013). Perceptual metrics for static and dynamic triangle meshes. *Computer Graphics Forum*, 32(1):101–125.

COSSAIRT, O., NAYAR, S. et RAMAMOORTHY, R. (2008). Light field transfer: global illumination between real and synthetic objects. *ACM Transactions on Graphics*, 27(3):57:1–57:6.

CURLISS, B. (1999). From range scans to 3d models. *SIGGRAPH Computer Graphics*, 33(4):38–41.

DANA, K., VAN-GINNEKEN, B., NAYAR, S. et KOENDERINK, J. (1999). Reflectance and Texture of Real World Surfaces. *ACM Transactions on Graphics (TOG)*, 18(1):1–34.

- DAVIS, A., LEVOY, M. et DURAND, F. (2012). Unstructured light fields. *Computer Graphics Forum*, 31(2pt1):305–314.
- DE FLORIANI, L., MAGILLO, P. et PUPPO, E. (1998). The MT (Multi-Tessellation) package. <http://www.disi.unige.it/person/MagilloP/MT/index.html>.
- DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W. et SAGAR, M. (2000). Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 145–156, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- DEBEVEC, P., YU, Y. et BOSHOKOV, G. (1998). Efficient view-dependent image-based rendering with projective texture-mapping. Rapport technique, University of California at Berkeley, Berkeley, CA, USA.
- DEBEVEC, P. E. (1996). *Modeling and Rendering Architecture from Photographs*. Thèse de doctorat, University of California at Berkeley, Computer Science Division, Berkeley CA.
- DO CARMO, M. P. (1976). *Differential Geometry of Curves and Surfaces*. Prentice-Hall.
- DUPUY, J., HEITZ, E., IEHL, J.-C., POULIN, P., NEYRET, F. et OSTROMOUKHOV, V. (2013). Linear efficient antialiased displacement and reflectance mapping. *ACM Transactions on Graphics*, 32(6):211:1–211:11.
- FILIP, J. et HAINDL, M. (2009). Bidirectional texture function modeling: A state of the art survey. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 31(11):1921–1940.
- FLEISHMAN, S., DRORI, I. et COHEN-OR, D. (2003). Bilateral mesh denoising. In *ACM SIGGRAPH 2003 Papers*, pages 950–953.
- FOO, S. (1997). *A Gonioreflectometer for Measuring the Bidirectional Reflectance of Material for Use in Illumination Computation*. Cornell University, Aug.
- FOOTE, J. et KIMBER, D. (2000). Flycam: practical panoramic video. In *Proceedings of the eighth ACM international conference on Multimedia*, MULTIMEDIA '00, pages 487–488, New York, NY, USA. ACM.
- FUCHS, H., BISHOP, G., ARTHUR, K., MCMILLAN, L., BISHOP, H. F. G., BAJCSY, R., LEE, S. W., FARID, H. et KANADE, T. (1994). Virtual space teleconferencing using a sea of cameras. In *Proceedings of the First International Conference on Medical Robotics and Computer Assisted Surgery*, pages 161–167.
- FUCHS, M., BLANZ, V., LENSCH, H. P. et SEIDEL, H.-P. (2007a). Adaptive sampling of reflectance fields. *ACM Transactions on Graphics*, 26(2).

- FUCHS, M., KÄCHELE, M. et RUSINKIEWICZ, S. (2013). Design and fabrication of faceted mirror arrays for light field capture. *Computer Graphics Forum*.
- FUCHS, M., LENSCH, H. P. A., BLANZ, V. et SEIDEL, H.-P. (2007b). Superresolution reflectance fields: Synthesizing images for intermediate light directions. *Computer Graphics Forum*, 26(3):447–456.
- FURUKAWA, Y. et PONCE, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 32(8):1362–1376.
- GARLAND, M. et HECKBERT, P. S. (1997). Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- GARLAND, M. et HECKBERT, P. S. (1998). Simplifying surfaces with color and texture using quadric error metrics. In *VIS '98: Proceedings of the conference on Visualization '98*, pages 263–269, Los Alamitos, CA, USA. IEEE Computer Society Press.
- GAUTRON, P., KRIVANEK, J., PATTANAIK, S. et BOUATOUCH, K. (2004). A novel hemispherical basis for accurate and efficient rendering. In *Proceedings of the Fifteenth Eurographics conference on Rendering Techniques*, EGSR'04, pages 321–330, Norrköping, Sweden. Eurographics Association.
- GÉNEVAUX, O. (2006). *Simulation de liquides à l'aide des équations de Navier-Stokes, et visualisation, à destination de l'infographie*. Thèse de doctorat, Université Louis Pasteur - Strasbourg I.
- GERSHO, A. et GRAY, R. M. (1992). *Vector quantization and signal compression*. Kluwer Academic Publishers, Dordrecht, the Netherlands.
- GERSHUN, A. (1939). The light field. *Journal of Mathematics and Physics*, 18:51–151. Titre original « Svetovoe Pole », publié à Moscou en 1936.
- GIMBUTAS, Z. et GREENGARD, L. (2009). Short note: A fast and stable method for rotating spherical harmonic expansions. *Journal of Computers and Physics*, 228(16):5621–5627.
- GIROD, B., CHANG, C.-L., RAMANATHAN, P. et ZHU, X. (2003). Light field compression using disparity-compensated lifting. In *Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 2*, ICME '03, pages 373–376, Washington, DC, USA. IEEE Computer Society.
- GONZÁLEZ, C., CASTELLÓ, P. et CHOVER, M. (2007). A texture-based metric extension for simplification methods. In *GRAPP (GM/R)*, pages 69–76.
- GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R. et COHEN, M. F. (1996). The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and*

- interactive techniques*, SIGGRAPH '96, pages 43–54, New York, USA. ACM.
- GOURAUD, H. (1971). Continuous shading of curved surfaces. *IEEE Transactions on Computers*, 20(6):623–629.
- GOURDON, A. (1995). Simplification of irregular surface meshes in 3d medical images. In *CVRMed*, volume 905 de *Lecture Notes in Computer Science*, pages 413–419. Springer.
- GUÉZIEC, A. (1995). Surface simplification with variable tolerance. In *Second annual international symposium on medical robotics and computer assisted surgery (MRCAS '95)*, pages 132–139.
- GUILD, J. (1932). The colorimetric properties of the spectrum. *Philosophical Transactions of the Royal Society*, 230(681-693):149–187.
- GUTHE, M., MÜLLER, G., SCHNEIDER, M. et KLEIN, R. (2009). Btf-cielab: A perceptual difference measure for quality assessment and compression of btfs. *Computer Graphics Forum*, 28(1):101–113.
- HAMANN, B. (1994). A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, 11(2):197–214.
- HAPKE, B. (2005). *Theory of Reflectance and Emittance Spectroscopy*. Topics in Remote Sensing. Cambridge University Press.
- HASTIE, T., TIBSHIRANI, R. et FRIEDMAN, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, USA.
- HAWKINS, T., COHEN, J. et DEBEVEC, P. (2001). A photometric approach to digitizing cultural artifacts. In *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, VAST '01, pages 333–342, New York, NY, USA. ACM.
- HAWKINS, T., EINARSSON, P. et DEBEVEC, P. (2005). A dual light stage. In *Proceedings of the 16th Eurographics conference on Rendering Techniques*, EGSR'05, pages 91–98, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- HE, X. D., TORRANCE, K. E., SILLION, F. X. et GREENBERG, D. P. (1991). P.: A comprehensive physical model for light reflection. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991*, pages 175–186.
- HECKBERT, P. et GARLAND, M. (1997). Survey of Polygonal Surface Simplification Algorithms. Rapport technique, Carnegie Mellon University, Pittsburgh, PA 15213.
- HEIGL, B., KOCH, R., POLLEFEYS, M., DENZLER, J. et VAN GOOL, L. J. (1999). Plenoptic modeling and rendering from image sequences taken by hand-held camera. In *Mustererkennung 1999, 21. DAGM-Symposium*, pages 94–101, London, UK, UK. Springer-Verlag.

HERZOG, R., CADIK, M., AYDCIN, T. O., KIM, K. I., MYSZKOWSKI, K. et SEIDEL, H.-P. (2012). Norm: No-reference image quality metric for realistic image synthesis. *Computer Graphics Forum*, 31(2pt4):545–554.

HOPPE, H. (1996). Progressive meshes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99–108, New York, NY, USA. ACM.

HOPPE, H. (1997). View-dependent refinement of progressive meshes. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 189–198, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

HOPPE, H. (1999). New quadric metric for simplifying meshes with appearance attributes. In *VISUALIZATION '99: Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, Washington, DC, USA. IEEE Computer Society.

HOPPE, H., DEROSE, T., DUCHAMP, T., McDONALD, J. et STUETZLE, W. (1993). Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 19–26, New York, NY, USA. ACM.

HOPPE, H. et MARSCHNER, S. (2000). Efficient minimization of new quadric metric for simplifying meshes with appearance attributes.

IHM, I., PARK, S. et LEE, R. K. (1997). Rendering of spherical light fields. In *Proceedings of the 5th Pacific Conference on Computer Graphics and Applications*, PG '97, pages 59–, Washington, DC, USA. IEEE Computer Society.

IMMEL, D. S., COHEN, M. F. et GREENBERG, D. P. (1986). A radiosity method for non-diffuse environments. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, pages 133–142, New York, NY, USA. ACM.

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE et CONSIGLIO NAZIONALE DELLE RICERCHE (2005). MeshLab. <http://meshlab.sourceforge.net/>.

ISAKSEN, A., McMILLAN, L. et GORTLER, S. J. (2000). Dynamically reparameterized light fields. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 297–306, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

JACHNIK, J., NEWCOMBE, R. A. et DAVISON, A. J. (2012). Real-time surface light-field capture for augmentation of planar specular surfaces. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, ISMAR '12, pages 91–97, Washington, DC, USA. IEEE Computer Society.

JENSEN, H. W., MARSCHNER, S. R., LEVOY, M. et HANRAHAN, P. (2001). A practical model for subsurface light transport. In *Proceedings of the 28th annual*

conference on Computer graphics and interactive techniques, SIGGRAPH '01, pages 511–518, New York, NY, USA. ACM.

KAJIYA, J. T. (1986). The rendering equation. *In Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, pages 143–150, New York, NY, USA. ACM.

KALVIN, A. D., CUTTING, C. B., HADDAD, B. et NOZ, M. E. (1991). Constructing topologically connected surfaces for the comprehensive analysis of 3-d medical structures. *In Proceedings of SPIE 1445, Medical Imaging V: Image Processing*.

KAZHDAN, M., BOLITHO, M. et HOPPE, H. (2006a). Poisson surface reconstruction. *In Proceedings of the 4th Eurographics symposium on Geometry processing*, pages 61–70.

KAZHDAN, M., BOLITHO, M. et HOPPE, H. (2006b). Screened poisson surface reconstruction (version 5.5).

KIM, C., ZIMMER, H., PRITCH, Y., SORKINE-HORNUNG, A. et GROSS, M. (2013). Scene reconstruction from high spatio-angular resolution light fields. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 32(4):73:1–73:12.

KIM, H. S., CHOI, H. K. et LEE, K. H. (2008). Mesh simplification with vertex color. *In GMP*, pages 258–271.

KITAHARA, M., SHIMIZU, S., KAMIKURA, K. et YOSHIYUKI, Y. (2005). Hardware accelerated image-based rendering with compressed surface light fields and multiresolution geometry. *In Proceedings of the 6th Pacific-Rim conference on Advances in Multimedia Information Processing - Volume Part I*, PCM'05, pages 96–107, Berlin, Heidelberg. Springer-Verlag.

KOHLER, J., NOLL, T., REIS, G. et STRICKER, D. (2013). A full-spherical device for simultaneous geometry and reflectance acquisition. *Applications of Computer Vision, IEEE Workshop on Applications of Computer Vision (WACV)*, pages 355–362.

KOVACS, P. T. et ZILLY, F. (2012). 3d capturing using multi-camera rigs, real-time depth estimation and depth-based content creation for multi-view and light-field auto-stereoscopic displays. *In ACM SIGGRAPH 2012 Emerging Technologies*, SIGGRAPH '12, pages 1:1–1:1, New York, NY, USA. ACM.

KRAEMER, P., UNTEREINER, L., JUND, T., THERY, S. et CAZIER, D. (2013). CGoGN: N-dimensional Meshes with Combinatorial Maps. *In Proceedings of the 22nd International Meshing Roundtable*. Springer Berlin Heidelberg.

LAFORTUNE, E. P. F., FOO, S.-C., TORRANCE, K. E. et GREENBERG, D. P. (1997). Non-linear approximation of reflectance functions. *In Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 117–126, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

- LAM, P.-M., LEUNG, C.-S. et WONG, T.-T. (2006). Noise-resistant fitting for spherical harmonics. *IEEE Transactions on Visualization and Computer Graphics*, 12:254–265.
- LAMBERT, J. H. (1760). *Photometria sive de mensura de gratibus luminis, colorum umbrae*. Eberhard Klett.
- LARUE, F. (2008). *Numérisation de pièces d'art en termes de forme et d'apparence pour la visualisation réaliste en synthèse d'images*. Thèse de doctorat, Université Louis Pasteur - Strasbourg I.
- LARUE, F. et DISCHLER, J.-M. (2006). Automatic registration and calibration for efficient surface light field acquisition. In *Proceedings of the 7th International conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage, VAST'06*, pages 171–178. Eurographics Association.
- LAVOUÉ, G., GELASCA, E. D., DUPONT, F., BASKURT, A. et EBRAHIMI, T. (2006). Perceptually driven 3d distance metrics with application to watermarking. In *SPIE proceedings*, volume 6312.
- LEBEDEV, V. (1976). Quadratures on a sphere. *USSR Computational Mathematics and Mathematical Physics*, 16(2):10 – 24.
- LELESCU, D. et BOSSEN, F. (2004). Representation and coding of light field data. *Graphics Models*, 66(4):203–225.
- LEONARDO DA VINCI (1500). The notebooks of leonardo da vinci. E-book online available at gutenberg.org.
- LEVOY, M. et HANRAHAN, P. (1996). Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42.
- LEVOY, M., NG, R., ADAMS, A., FOOTER, M. et HOROWITZ, M. (2006). Light field microscopy. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 924–934, New York, NY, USA. ACM.
- LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J. et FULK, D. (2000). The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 131–144, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- LI, H., FOO, S. C., TORRANCE, K. E. et WESTIN, S. H. (2005). Automated three-axis gonireflectometer for computer graphics applications. In *Optical Engineering*, page 2006.

- LIPPMANN, G. (1908). Épreuves réversibles donnant la sensation du relief. *Journal of Physics and Theoretical Applications*, 7.
- LOW, K.-L. et TAN, T.-S. (1997). Model simplification using vertex-clustering. *In Proceedings of the 1997 symposium on Interactive 3D graphics, I3D '97*, pages 75–ff., New York, NY, USA. ACM.
- LYTRO INC. (2010). <http://www.lytro.com/>.
- MACROBERT, T. M. (1948). *Spherical harmonics: an elementary treatise on harmonic functions, with applications*. Dover Publications.
- MALZBENDER, T., GELB, D. et WOLTERS, H. (2001). Polynomial texture maps. *In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pages 519–528, New York, NY, USA. ACM.
- MANN, S. et PICARD, R. W. (1994). Virtual bellows: Constructing high quality stills from video. *In ICIP (1)*, pages 363–367. IEEE.
- MARQUES, R., BOUVILLE, C., RIBARDIÈRE, M., SANTOS, L. P. et BOUATOUCH, K. (2013). Spherical fibonacci point sets for illumination integrals. *Computer Graphics Forum*, 32(8):134–143.
- MASHIKO, T., YAGOU, H., WEI, D., DING, Y. et WU, G. (2004). 3d triangle mesh smoothing via adaptive mmse filtering. *In 2004 International Conference on Computer and Information Technology (CIT 2004), 14-16 September 2004, Wuhan, China*, pages 734–740.
- MCAULEY, S., HILL, S., HOFFMAN, N., GOTANDA, Y., SMITS, B., BURLEY, B. et MARTINEZ, A. (2012). Practical physically-based shading in film and game production. *In ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 10:1–10:7, New York, NY, USA. ACM.
- MCMILLAN, L. et BISHOP, G. (1995). Plenoptic modeling: an image-based rendering system. *In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 39–46, New York, NY, USA. ACM.
- METROPOLIS, N. et ULAM, S. (1949). The monte carlo method. *Journal of the American statistical Association*, 44(247):335–341.
- MILLER, G. S. P., RUBIN, S. M. et PONCELEON, D. B. (1998). Lazy decomposition of surface light fields for precomputed global illumination. *In Rendering Techniques*, pages 281–292. Springer.
- MITTAL, A., SOFAT, S. et HANCOCK, E. (2012). An efficient scheme for color edge detection in uniform color space. *In Autonomous and Intelligent Systems, Lecture Notes in Computer Science*, pages 260–267. Springer Berlin Heidelberg.

- MÜLLER, G., BENDELS, G. H. et KLEIN, R. (2005). Rapid synchronous acquisition of geometry and appearance of cultural heritage artefacts. *In Proceedings of the 6th International conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage, VAST'05*, pages 13–20, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- NALWA, V. S. (1996). A true omnidirectional viewer. Rapport technique, Bell laboratories.
- NAYAR, S. K. (1997). Catadioptric omnidirectional camera. *In Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 482–, Washington, DC, USA. IEEE Computer Society.
- NAYAR, S. K., IKEUCHI, K. et KANADE, T. (1991). Surface reflection: Physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:611–634.
- NEUMANN, U., PINTARIC, T. et RIZZO, A. (2000). Immersive panoramic video. *In Proceedings of the eighth ACM international conference on Multimedia, MULTIMEDIA '00*, pages 493–494, New York, NY, USA. ACM.
- NICODEMUS, F. E. (1965). Directional reflectance and emissivity of an opaque surface. *Applied Optics*, 4(7):767–773.
- NICODEMUS, F. E., RICHMOND, J. C., HSIA, J. J., GINSBERG, I. W. et LIMPERIS, T. (1992). Geometrical considerations and nomenclature for reflectance. *In Radiometry*, chapitre Geometrical considerations and nomenclature for reflectance, pages 94–145. Jones and Bartlett Publishers, Inc., USA.
- NISHINO, K. et IKEUCHI, K. (2002). Robust simultaneous registration of multiple range images. *In Proceedings of the 5th Asian Conference on Computer Vision*, pages 454–461.
- NISHINO, K., SATO, Y. et IKEUCHI, K. (2001). Eigen-texture method: Appearance compression and synthesis based on a 3d model. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 23(11):1257–1265.
- OREN, M. et NAYAR, S. K. (1994). Generalization of lambert's reflectance model. *In Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH '94*, pages 239–246, New York, NY, USA. ACM.
- PHONG, B. T. (1975). Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317.
- PIGHIN, F., HECKER, J., LISCHINSKI, D., SZELISKI, R. et SALESIN, D. H. (1998). Synthesizing realistic facial expressions from photographs. *In Proceedings of the 25th annual conference on Computer graphics and interactive techniques, SIGGRAPH '98*, pages 75–84, New York, NY, USA. ACM.

- POYNTON, C. (1997). Frequently Asked Questions about Color. www.poynton.com.
- PULLI, K., COHEN, M., DUCHAMP, T., HOPPE, H., SHAPIRO, L. et STUETZLE, W. (1997). View-based rendering: Visualizing real objects from scanned range and color data. In *Eurographics Rendering Workshop*, pages 23–34.
- RAMAMOORTHY, R. et BIGGS, E. (2009). Representation of visual appearance. Course in advanced computer graphics.
- ROSSIGNAC, J. et BORREL, P. (1993). Multi-resolution 3d approximations for rendering complex scenes. In *Modeling in Computer Graphics*, IFIP Series on Computer Graphics, pages 455–465. Springer Berlin Heidelberg.
- RUITERS, R., SCHWARTZ, C. et KLEIN, R. (2012). Data driven surface reflectance from sparse and irregular samples. *Computer Graphics Forum*, 31(2pt1):315–324.
- RUSINKIEWICZ, S. et LEVOY, M. (2001). Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*.
- SALVI, J., FERNANDEZ, S., PRIBANIC, T. et LLADO, X. (2010). A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, 43(8):2666 – 2680.
- SANDER, P. V., SNYDER, J., GORTLER, S. J. et HOPPE, H. (2001). Texture mapping progressive meshes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA. ACM.
- SCHIRMACHER, H. (2003). *Efficient Acquisition, Representation, and Rendering of Light Fields*. Thèse de doctorat, Universität des Saarlandes, Saarbrücken.
- SCHMITT, E. (2013). Diffusion sur des maillages pour la reconstruction de champs de lumière.
- SCHRÖDER, P. et SWELDENS, W. (1995). Spherical wavelets: efficiently representing functions on the sphere. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 161–172, New York, NY, USA. ACM.
- SCHROEDER, W. J., ZARGE, J. A. et LORENSEN, W. E. (1992). Decimation of triangle meshes. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, pages 65–70, New York, NY, USA. ACM.
- SCHWARTZ, C., RUITERS, R. et KLEIN, R. (2013). Level-of-detail streaming and rendering using bidirectional sparse virtual texture functions. *Computer Graphics Forum (Proceedings of Pacific Graphics)*, 32(7):345–354.

- SCOPIGNO, R., CALLIERI, M., CIGNONI, P., CORSINI, M., DELLEPIANE, M., PONCHIO, F. et RANZUGLIA, G. (2011). 3d models for cultural heritage: Beyond plain visualization. *Computer*, 44(7):48–55.
- SHUM, H. et KANG, S. (2000). A review of image-based rendering techniques. *In VCIP00*, pages 2–13.
- SHUM, H.-Y., CHAN, S.-C. et KANG, S. B. (2006). *Image-Based Rendering*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- SHUM, H.-Y. et HE, L.-W. (1999). Rendering with concentric mosaics. *In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99*, pages 299–306, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- SHUM, H.-Y., KANG, S. B. et CHAN, S.-C. (2003). Survey of image-based representations and compression techniques. *IEEE Transactions on Circuits and Systems for Video Techniques*, 13(11):1020–1037.
- SLOAN, P.-P., COHEN, M. F. et GORTLER, S. J. (1997). Time critical lumigraph rendering. *In Proceedings of the 1997 symposium on Interactive 3D graphics, I3D '97*, pages 17–ff., New York, NY, USA. ACM.
- STOKES, G. G. (1883). *On the perfect Blackness of the Central Spot in Newton's Rings, and on the Verification of Fresnel's Formula for the intensities of Reflected and Reflected Rays*. Cambridge University Press.
- SUBSOL, G., MORENO, B., JESSEL, J.-P., BRUXELLES, L., BRAGA, J., CLARKE, R. et THACKERAY, F. (2011). 3d digitization of the excavation site of a fossil hominid (stw 573 / "little foot", sterkfontein, south africa (regular paper). *In Paleoanthropology Society Annual Meeting, Minneapolis (U.S.A.), 12/04/2011-13/04/2011*, page (electronic medium), <http://www.paleoanthro.org/>. Paleoanthropology Society.
- SZELISKI, R. (1996). Video mosaics for virtual environments. *IEEE Computer Graphics Applications*, 16(2):22–30.
- SZELISKI, R. et SHUM, H.-Y. (1997). Creating full view panoramic image mosaics and environment maps. *In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97*, pages 251–258, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- TAN, P., LIN, S., QUAN, L., GUO, B. et SHUM, H. (2008). Filtering and rendering of resolution-dependent reflectance models. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):412–425.
- THOMAS, J.-B., COLANTONI, P. et TRÉMEAU, A. (2013). On the uniform sampling of cielab color space and the number of discernible colors. *In Proceedings of the 4th international conference on Computational Color Imaging, CCIW'13*, pages 53–67, Berlin, Heidelberg. Springer-Verlag.

- TODT, S., REZK-SALAMA, C., KOLB, A. et KUHNERT, K.-D. (2008). Gpu-based spherical light field rendering with per-fragment depth correction. *Computer Graphics Forum*, 27(8):2081–2095.
- TOMPkin, J., MUFF, S., JAKUSCHEVSKIJ, S., MCCANN, J., KAUTZ, J., ALEXA, M. et MATUSIK, W. (2012). Interactive light field painting. In *ACM SIGGRAPH 2012 Emerging Technologies*, SIGGRAPH '12, pages 12:1–12:1, New York, NY, USA. ACM.
- TORRANCE, K. E. et SPARROW, E. M. (1992). Theory for off-specular reflection from roughened surfaces. In *Radiometry*, chapitre Theory for off-specular reflection from roughened surfaces, pages 32–41. Jones and Bartlett Publishers, Inc., USA.
- TSAI, R. Y. (1992). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. In *Radiometry*, chapitre A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, pages 221–244. Jones and Bartlett Publishers, Inc., USA.
- TURK, G. (1992). Re-tiling polygonal surfaces. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, pages 55–64, New York, NY, USA. ACM.
- VANHOEY, K., SAUVAGE, B. et DISCHLER, J.-M. (2010). Simplification de maillages surfaciques avec champs de lumière. *23èmes Journées de l'Association Française d'Informatique Graphique*.
- VANHOEY, K., SAUVAGE, B., GÉNEVAUX, O., LARUE, F. et DISCHLER, J.-M. (2013a). Robust fitting on poorly sampled data for surface light field rendering and image relighting. *Computer Graphics Forum*, 32(6):101–112.
- VANHOEY, K., SAUVAGE, B., LARUE, F. et DISCHLER, J.-M. (2013b). On-the-fly multi-scale infinite texturing from example. *ACM Trans. Graph.*, 32(6):208:1–208:10.
- ČADÍK, M., HERZOG, R., MANTIUK, R., MYszKOWSKI, K. et SEIDEL, H.-P. (2012). New measurements reveal weaknesses of image quality metrics in evaluating graphics artifacts. *ACM Transactions on Graphics*, 31(6):147:1–147:10.
- VEDULA, S., BAKER, S., RANDEr, P., COLLINS, R. et KANADE, T. (2005). Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 27(3):475–480.
- von HELMHOLTZ, H. (1867). *Handbuch der physiologischen Optik*. Leopold Voss, Leipzig.
- WAHBA, G. (1981). Spline Interpolation and Smoothing on the Sphere. *Siam Journal on Scientific and Statistical Computing*, 2:5–16.

- WAHBA, G. (1990). *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics.
- WALT DISNEY ANIMATION STUDIOS (2012). BRDF explorer.
- WARD, G. J. (1992). Measuring and modeling anisotropic reflection. *In Proceedings of the 19th annual conference on Computer graphics and interactive techniques, SIGGRAPH '92*, pages 265–272, New York, NY, USA. ACM.
- WILBURN, B., SMULSKI, M., LEE, K. et HOROWITZ, M. A. (2002). The light field video camera. *In Media Processors 2002*, pages 29–36.
- WILLIAMS, L. (1983). Pyramidal parametrics. *SIGGRAPH Computers and Graphics*, 17(3):1–11.
- WOOD, D. N., AZUMA, D. I., ALDINGER, K., CURLESS, B., DUCHAMP, T., SALESIN, D. H. et STUETZLE, W. (2000). Surface light fields for 3d photography. *In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*, pages 287–296, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- WRIGHT, W. D. (1929). A re-determination of the trichromatic coefficients of the spectral colours. *Transactions of the Optical Society*, 30(4):141.
- WYSZECKI, G. et STILES, W. S. (2000). *Color Science: Concepts and Methods, Quantitative Data and Formulae (Wiley Series in Pure and Applied Optics)*. Wiley-Interscience, 2 édition.
- XIONG, Y. et TURKOWSKI, K. (1997). Creating image-based vr using a self-calibrating fisheye lens. *In Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 237–243, Washington, DC, USA. IEEE Computer Society.
- YANG, J. C., EVERETT, M., BUEHLER, C. et McMILLAN, L. (2002). A real-time distributed light field camera. *In Proceedings of the 13th Eurographics workshop on Rendering, EGRW '02*, pages 77–86, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- ZHANG, C. et CHEN, T. (2003). Spectral analysis for sampling image-based rendering data. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1038–1050.
- ZHANG, C. et CHEN, T. (2004). A survey on image-based rendering—representation, sampling and compression. *Signal Processing: Image Communication*, 19(1):1 – 28.

Traitement conjoint de la géométrie et de la radiance d'objets 3D numérisés

Depuis quelques décennies, les communautés d'informatique graphique et de vision ont contribué à l'émergence de technologies permettant la numérisation d'objets 3D. Une demande grandissante pour ces technologies vient des acteurs de la culture, notamment pour l'archivage, l'étude à distance et la restauration d'objets du patrimoine culturel : statuettes, grottes et bâtiments par exemple. En plus de la géométrie, il peut être intéressant de numériser la photométrie avec plus ou moins de détail : simple texture (2D), champ de lumière (4D), SV-BRDF (6D), etc. Nous formulons des solutions concrètes pour la création et le traitement de champs de lumière surfaciques représentés par des fonctions de radiance attachés à la surface. Nous traitons le problème de la phase de construction de ces fonctions à partir de plusieurs prises de vue de l'objet dans des conditions « sur site » : échantillonnage non structuré voire peu dense et bruité. Un procédé permettant une reconstruction robuste générant un champ de lumière surfacique variant de « prévisible » et sans artefacts à excellente, notamment en fonction des conditions d'échantillonnage, est proposé. Ensuite, nous suggérons un algorithme de simplification permettant de réduire la complexité mémoire et calculatoire de ces modèles parfois lourds. Pour cela, nous introduisons une métrique qui mesure conjointement la dégradation de la géométrie et de la radiance. Finalement, un algorithme d'interpolation de fonctions de radiance est proposé afin de servir une visualisation lisse et naturelle, peu sensible à la densité spatiale des fonctions. Cette visualisation est particulièrement bénéfique lorsque le modèle est simplifié.

Mots clés : rendu basé image, numérisation 3D, patrimoine culturel, apparence, reconstruction, niveau de détail, simplification, rendu, radiance, luminance, champ de lumière, interpolation surfacique

Joint Treatment of Geometry and Radiance for 3D Model Digitisation

Vision and computer graphics communities have built methods for digitizing, processing and rendering 3D objects. There is an increasing demand coming from cultural communities for these technologies, especially for archiving, remote studying and restoring cultural artefacts like statues, buildings or caves. Besides digitizing geometry, there can be a demand for recovering the photometry with more or less complexity : simple textures (2D), light fields (4D), SV-BRDF (6D), etc. In this thesis, we present steady solutions for constructing and treating surface light fields represented by hemispherical radiance functions attached to the surface in real-world on-site conditions. First, we tackle the algorithmic reconstruction-phase of defining these functions based on photographic acquisitions from several viewpoints in real-world "on-site" conditions. That is, the photographic sampling may be unstructured and very sparse or noisy. We propose a process for deducing functions in a manner that is robust and generates a surface light field that may vary from "expected" and artefact-less to high quality, depending on the uncontrolled conditions. Secondly, a mesh simplification algorithm is guided by a new metric that measures quality loss both in terms of geometry and radiance. Finally, we propose a GPU-compatible radiance interpolation algorithm that allows for coherent radiance interpolation over the mesh. This generates a smooth visualisation of the surface light field, even for poorly tessellated meshes. This is particularly suited for very simplified models.

Keywords: digitization, cultural heritage, aspect, light field, luminance, radiance, acquisition, reconstruction, visualisation, simplification.