



HAL
open science

Apparence multi-échelles pour le rendu réaliste et efficace des surfaces complexes

Eric Heitz

► **To cite this version:**

Eric Heitz. Apparence multi-échelles pour le rendu réaliste et efficace des surfaces complexes. Modélisation et simulation. Université de Grenoble, 2014. Français. NNT: . tel-01073518v1

HAL Id: tel-01073518

<https://theses.hal.science/tel-01073518v1>

Submitted on 21 Oct 2014 (v1), last revised 22 May 2017 (v2)

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

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

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques Appliquées, Informatique**

Arrêté ministériel :

Présentée par

Eric Heitz

Thèse dirigée par **Fabrice Neyret**

préparée au sein **du Laboratoire Jean Kuntzmann**
et de **l'Ecole Doctorale Mathématiques, Sciences et Technologies de**
l'Information, Informatique de Grenoble

Apparence multi-échelles pour le rendu réaliste et efficace des surfaces complexes

Thèse soutenue publiquement le **26 septembre 2014**,
devant le jury composé de :

Mathias Paulin

Professeur à l'Université Paul Sabatier, Toulouse, Rapporteur

Jaroslav Křivánek

Professeur à Charles University, Prague, Rapporteur

François Sillion

Directeur de Recherche, INRIA, Grenoble, Examineur

Christophe Bourlier

Directeur de Recherche, CNRS, Nantes, Examineur

Jos Stam

Research Scientist, Alias Wavefront, Toronto, Examineur

Sébastien Lagarde

Senior Graphic Programmer, DICE, Stockholm, Examineur

Fabrice Neyret

Directeur de Recherche, CNRS, Grenoble, Directeur de thèse



Remerciements

Mes premiers remerciements vont à Fabrice Neyret, mon directeur de thèse, dont la contribution à l'élaboration de ce travail et à la construction de mon identité de chercheur dépasse largement les limites d'un simple encadrement scientifique. Fabrice a partagé avec moi une vision exceptionnelle de la science que je n'ai décelée chez nul autre et qui m'a profondément influencé. L'impact en est perceptible dans la totalité des projets que j'ai réalisés pendant mes années de thèse – y compris ceux sur lesquels j'ai travaillé seul ou avec d'autres collaborateurs – et il continuera d'imprégner durablement mon devenir de chercheur. Je remercie aussi Fabrice pour son exigence qui m'a épargné de tomber dans la facilité. Durant toute ma scolarité, les mathématiques et l'informatique ont toujours été faciles et n'ont jamais requis de ma part beaucoup d'efforts pour réussir. Mais la rédaction d'un article ou la présentation de résultats de travaux demande des compétences autres que je n'avais pas et dont l'acquisition – avouons-le – ne s'est pas faite de manière indolore. Alors que j'ai opposé à Fabrice une résistance non négligeable et qu'il m'est arrivé de montrer une certaine animosité, il a su faire preuve de patience, de tolérance et a su me convaincre de fournir les efforts requis pour surmonter mes réticences : “Quand ça fait mal, c'est que le muscle a besoin de travailler”, me disait-il. J'ai détesté cette phrase. Mais aujourd'hui, j'ai conscience des progrès que je lui dois : si j'ai, à plusieurs reprises, obtenu le prix du meilleur papier en conférence, c'est grâce aux efforts que j'ai déployés pour répondre aux exigences de mon directeur de thèse. Enfin, il me faut exprimer aussi à Fabrice ma profonde reconnaissance pour ses qualités humaines : sa grande proximité, son exemplaire écoute et sa sincère implication dans l'encadrement qu'il a assumé.

Jonathan Dupuy est la deuxième personne avec laquelle j'ai le plus collaboré. Jonathan a commencé sa thèse un an après moi et travaille également sur le problème du filtrage de l'apparence des surfaces, ce qui pouvait nous placer dans une situation soit de concurrence, soit de collaboration. Nous avons choisi la seconde option, mais cela n'a pas toujours été facile à gérer (surtout pour moi). Cependant, nous avons réussi – et nous continuons – à collaborer sur des projets d'une qualité qui nous serait sans doute difficilement accessible si chacun faisait cavalier seul. Contrairement aux expériences que j'ai vécues dans le cadre d'autres projets, dans la collaboration avec Jonathan, il m'est arrivé d'avoir l'impression de vivre des situations où c'est lui qui apportait des idées audacieuses et qui explorait avec succès des pistes qui semblaient a priori ne mener nulle part, tandis que mon rôle consistait plutôt à suivre la piste en mettant les bons jacobiniens aux bons endroits et à apporter de la rigueur méthodologique au projet. Jonathan reste mon interlocuteur privilégié lorsqu'il s'agit d'avoir des conversations pointues sur les micro-facettes. Merci Jonathan, je suis très fier des projets sur lesquels nous avons travaillé ensemble : je n'aurais pu les réaliser sans toi.

Je suis redevable à de nombreuses personnes pour l'ensemble des projets associés à ma thèse. Merci à Sébastien Lagarde de m'avoir encouragé à écrire mon rapport de recherche sur la fonction de masquage. Finalement, ce rapport est devenu l'article probablement le plus important de ma thèse et m'a valu d'être invité à participer au cours SIGGRAPH sur le physically based shading. Je remercie aussi Stephen Hill qui a relu ce document de soixante pages, et ce, un nombre de fois que je n'ose pas compter : il a notamment corrigé les fautes d'anglais qui émaillaient le document. Merci aussi à Naty Hoffman et à Morgan McGuire pour leurs précieux retours. Merci à Christophe Bourlier et à Nicolas Pinel qui ont accepté de m'aider à publier un article dans la communauté de la physique dont les codes sont différents de ceux de l'informatique graphique. Je n'avais pas les billes pour le faire tout seul et j'ai beaucoup appris à travers cette expérience. Merci à Eugene d'Eon d'avoir collaboré avec moi sur notre projet d'importance sampling.

Ma thèse a aussi été l'occasion de voyager et de découvrir de nouveaux horizons. Je remercie Pierre Poulin, Derek Nowrouzezahrai et les autres du LIGUM pour m'avoir intégré à leur équipe pendant six mois. Merci spécialement à Pierre pour son accueil très chaleureux. Je remercie aussi Cyril Crassin et ses collègues du bureau NVIDIA à Paris ; je garde un superbe souvenir de cette expérience. Et merci à Antoine Bouthors et ses collègues de Weta Digital pour leur invitation. Passer de l'autre côté de la barrière et découvrir le côté industriel de la force a été très enrichissant pour moi. N'ayant malheureusement pas pu y séjourner bien longtemps, je me réjouirais de pouvoir y retourner un jour !

Cette phase d'études doctorales a été une expérience des plus enrichissantes qu'il m'ait été donné de vivre. La motivation nécessaire à la réalisation de cette thèse a été puissamment stimulée par l'ambiance exceptionnelle qui règne dans l'équipe Artis/Maverick. A toutes les personnes qui ont fait mon quotidien pendant ces quatre dernières années j'exprime ma sincère gratitude. Merci à Laurent d'avoir joué avec moi au ping-pong pendant les deux premières années et de m'avoir abreuvé en informations juteuses au quotidien sur Gmail les deux années suivantes. Merci à Morgan pour son humour noir, nos conversations techniques, culturelles, sociétales et, bien sûr, notre projet d'électronique. Merci à Hugo pour les nombreuses nuits blanches que nous avons passées ensemble au labo à travailler, à discuter de tout et de rien et à explorer les diverses ressources nocturnes de l'INRIA. Elles font partie de mes souvenirs de thèse les plus mémorables. Merci surtout pour la dernière nuit, la veille de ma soutenance, que nous avons passée à faire des répétitions au prix d'une lutte acharnée contre le sommeil. Merci à Benoit d'avoir su s'approprier la culture du troll dans l'équipe et d'avoir su en repousser les limites. Merci à JJ d'avoir rehaussé le niveau technique de nos conversations pendant les moments d'égarement. Merci à Csol pour la complicité qu'il continue d'entretenir avec les thésards et qui est loin d'être courante chez les permanents. Mais bon... "On ne peut pas sauver tout le monde!". Merci à JDG, toujours bon public pour les trolls à la pause, pour le chocolat. Merci à Charles de Rousiers pour ses conseils judicieux en début de thèse et pour son script vidéo. Merci à Pierre Bénard pour ses gâteaux et son sens de l'humour très british. Merci aussi à Jonathan El Methni et Gildas Mazo, mes amis de chez Mistis, pour les parties de babyfoot et les énigmes mathématiques.

Merci à tous ceux qui se sont déplacés pour être présents à ma soutenance, amis et famille. Merci aux membres du jury d'être venus et, surtout, de s'être infligé la lecture de ce manuscrit de 285 pages.

Et enfin, merci à Sabrina, qui a été à mes côtés et m'a encouragé au quotidien tout au long de mes années doctorales et qui a supporté mes absences prolongées ou parfois, simplement, mes indisponibilités. Merci pour ton soutien inconditionnel et pour ta patience dans les moments les plus difficiles.

Résumé

Le rendu, à la fois réaliste et efficace des scènes complexes, est un défi majeur de l'image de synthèse. Dans le domaine du rendu photo-réaliste, l'image est calculée en simulant les interactions physiques entre la lumière et le contenu de la scène. Mais, aujourd'hui, les studios de production ne créent plus, pour leurs films, de simples scènes, mais des mondes entiers, à la fois gigantesques et détaillés. La complexité s'est accrue de manière exponentielle, entraînant un problème de scalabilité. La masse de données et la complexité des interactions sont telles que le calcul de l'image est devenu déraisonnablement coûteux, et ce, même sur des serveurs de calcul très puissants. Dans le domaine du rendu temps-réel, l'objectif de photo-réalisme est moins poussé. Le problème posé par la complexité est alors celui de la quantité de détails sous-pixel qui sont source d'artefacts visuels comme l'aliasing, le popping, et l'incohérence de l'apparence avec les changements de distance.

Les détails sous-pixel ont une influence sur l'apparence au niveau du pixel et ne peuvent donc être simplement omis. De fait, réaliser un rendu multi-échelles en traitant séparément les détails vus de loin comme des matériaux est une solution naturelle à ce problème. Cependant, les modèles multi-échelles existants, souvent empiriques ou analysés avec force approximations, sont peu précis ou, du moins, n'ont pas été conçus pour représenter des situations complexes et échouent à préserver l'apparence des objets et donc à garantir la qualité photo-réaliste de l'image, ce qui les rend inutilisables avec les exigences de qualité en production. Nous soutenons la thèse qu'il est possible, en passant par une modélisation rigoureuse du problème de l'apparence au niveau des pixels, de mettre au point des algorithmes de rendu multi-échelles efficaces qui calculent des images de qualité. Nous explorons cette idée dans le cas des surfaces détaillées pour lesquelles nous utilisons et revisitons la théorie des micro-facettes comme fond théorique.

La première partie de ce mémoire est dédiée à la mise au point du modèle physique à micro-facettes qui décrit les interactions entre la lumière et une surface multi-échelles. Nous commençons par revisiter les fondements de la théorie des micro-facettes. En partant de la conservation de l'aire projetée visible des micro-facettes, nous redérivons et unifions les principaux modèles existants. Ensuite, nous étendons la théorie pour la rendre compatible avec un paradigme de rendu multi-échelles. La compréhension acquise sur le modèle nous permet aussi de dériver de nouveaux résultats. En particulier, nous montrons comment certaines propriétés d'invariance permettent de dériver simplement des résultats pour des micro-surfaces anisotropes. Enfin, nous proposons une métrique de validation pour les modèles à micro-facettes, que nous utilisons pour valider les résultats existants ainsi que les nouveaux que nous proposons.

Dans la seconde partie, nous montrons comment utiliser ces résultats théoriques pour mettre au point des applications de rendu de surfaces multi-échelles. En pratique, il faut choisir une représentation pour les surfaces multi-échelles et leurs matériaux, et calculer leurs interactions avec l'éclairage. Nous utilisons une représentation mémoire simple et légère pour les surfaces et montrons comment la rendre compatible avec l'animation des modèles, afin de ne pas avoir à la recalculer pour chaque image d'une séquence animée. Lorsque les attributs géométriques et optiques du matériau sont corrélés, ils doivent être représentés et évalués d'une manière qui dépend de leur visibilité, à la fois pour la caméra et pour la lumière. Nous montrons que ce problème peut toujours s'exprimer comme un problème de color mapping dont nous dérivons la solution ainsi qu'une manière efficace de la calculer. Enfin, nous décrivons des algorithmes d'intégration numérique de l'éclairage incident dédiés au rendu temps-réel et offline.

Abstract

Efficient rendering of realistic and complex scenes is a major challenge for image synthesis. In the field of photorealistic rendering, the image is computed by simulating the physical interactions between the light and the content of the scene. Nowadays, production studios don't create simple scenes for their movies, but gigantic and extremely detailed worlds. This exponential rise of complexity is responsible for scalability problems. The amount of data and the complexity of the interactions are such that computing the image has become unreasonably costly, and this even on the most powerful computers. In the field of real-time rendering, the objective of photorealism is less pushed on, but the problem posed by the complexity resides in the amount of subpixel details. They are the source of artefacts like aliasing, popping and inconsistent changes of appearances.

Subpixel details influence appearance at the level of the pixel and cannot just be removed. Indeed, the transition from pixel to subpixel during a zoom requires recovering the emerging visual effects with high accuracy. Hence, using a multi-scale strategy, in which subpixel details are represented like surface material, is a natural solution to this scalability problem. However, existing multi-scale models are mostly empirical and approximate. They are not designed to handle complex scenes and fail at recovering the correct appearance; they don't satisfy the requirements for photorealistic quality demanded in production. We claim that it is possible to design efficient and high-quality multi-scale rendering algorithms. This can be done by modeling rigorously the problem of filtering the appearance at the level of the pixels. We explore this idea in the case of detailed surfaces, for which we use and revisit the theoretical background provided by microfacet theory.

In the first part of this manuscript, we design the physical microfacet-based model, which describes the interactions between light and a multi-scale surface. We start by revisiting the foundations of microfacet theory. Starting from the conservation of the visible projected area, we derive and unify existing results. Then, we extend the theory and make it compatible with multi-scale rendering paradigms. The understanding acquired from the model also allows us to derive new results. For instance, we show how invariance properties can be used to derive results in the case of anisotropic microspheres, for which anisotropy is due to stretched micro-geometry, the typical case which occurs with animated surfaces. We also propose a metric that we use to validate microfacet-based models.

In the second part, we show how to use these theoretical results to design multi-scale rendering applications for detailed surfaces. In practice, one needs to choose a representation for multi-scale surfaces and their materials, and compute how they interact with the incident light. We use a lightweight memory representation and show how to make it compatible with the animation of the models, so that there is no need to initialize it again for each image of an animated sequence. When the geometrical and optical attributes of the material are correlated, they have to be represented and evaluated in a way that depends on their visibility for the camera and the light source directions. We show that this problem can always be expressed as a color mapping problem. We derive its solution and an efficient way to evaluate it. Finally, we describe numerical integration schemes for the incident light dedicated to real-time and offline rendering. In particular, we propose a new importance sampling technique for microfacet-based materials that can be used in a typical Monte Carlo renderer.

Table des matières

1	Introduction	13
1.1	La complexité débordante des scènes en synthèse d'images	13
1.1.1	La synthèse d'images	13
1.1.2	Le problème du coût de la complexité	13
1.1.3	Objectif de la thèse	15
1.2	Approche générale	16
1.2.1	Le rendu est un problème d'intégration	16
1.2.2	La limite des approches "brute force"	17
1.2.3	Le modèle du rendu filtré	18
1.2.4	Les fausses pathologies du rendu filtré	20
1.2.5	Vers un modèle de rendu filtré bien posé	23
1.3	Périmètre thématique	24
1.3.1	Critères pour choisir un modèle d'étude	24
1.3.2	Les détails surfaciques comme primitives multi-échelles	25
1.3.3	Objectif formel de la thèse	31
1.3.4	Description formelle de la thèse	32
1.3.5	Organisation des chapitres	35
1.3.6	Organisation des annexes	36
1.3.7	Communications associées à la thèse	38
1.3.8	Collaborations académiques et industrielles associées à la thèse	39
1.3.9	Impact mesurable à court terme des travaux associés à la thèse	40
I	Revisiter la théorie des micro-facettes	41
2	Rappels généraux sur les BRDFs	43
2.1	Quantités physiques	43
2.2	Configuration géométrique	44
2.3	Définition	44
2.4	Modèles idéaux	45
2.5	Modèles empiriques	46
3	Rappels sur la théorie des micro-facettes	49
3.1	Le modèle statistique de micro-surface	49
3.1.1	La distribution de normales D	49
3.1.2	La fonction de masquage G_1	50
3.1.3	La fonction de masquage et d'ombrage G_2	51
3.2	Les modèles physiques de matériaux	53
3.2.1	Les matériaux diffractants	53
3.2.2	Le modèle à micro-facettes spéculaires	53

3.2.3	Le modèle à micro-facettes diffuses	54
3.2.4	Les modèles multi-couches	54
4	Comprendre la fonction de masquage et d'ombrage dans les modèles à micro-facettes	55
4.1	Questions ouvertes dans la littérature	56
4.1.1	Sur le choix de la fonction de masquage	56
4.1.2	Trouver des critères quantitatifs	57
4.2	Dérivation de la fonction de masquage	60
4.2.1	Mesure de la radiance sur une surface	60
4.2.2	Statistiques des micro-facettes	61
4.2.3	Projections des micro-facettes	64
4.2.4	Une contrainte sur la fonction de masquage	64
4.2.5	Résumé	65
4.3	Les modèles de BRDFs à micro-facettes	66
4.3.1	La distribution de normales visibles	66
4.3.2	Construction d'une BRDF à micro-facettes	67
4.3.3	Construction d'une BRDF à micro-facettes spéculaires	68
4.3.4	Construction d'une BRDF à micro-facettes diffuses	68
4.3.5	Le test de normalisation de la BRDF	69
4.3.6	Résumé	70
4.4	Choisir un profil de micro-surface	72
4.4.1	Le profil de micro-surface de Smith	72
4.4.2	Le profil de micro-surface à V-cavités	75
4.4.3	Les fonctions de masquages non physiques	80
4.4.4	Résumé	82
4.5	Invariance d'étirement de la fonction de masquage	84
4.5.1	Invariance de la probabilité de masquage	84
4.5.2	La distribution des pentes	84
4.5.3	Distributions de pentes isotropes à forme invariante	85
4.5.4	Distributions de pentes anisotropes à forme invariante	86
4.5.5	Plus de généralisation	89
5	Théorie multi-échelles des micro-facettes	91
5.1	Introduction de l'échelle mésoscopique	92
5.1.1	Motivation	92
5.1.2	Définition de l'échelle mésoscopique	92
5.1.3	Les approches précédentes	93
5.2	Redériver les résultats précédents pour des distributions décentrées	95
5.2.1	Implication du décentrage pour le modèle de BRDF	95
5.2.2	Conservation de l'aire projetée de la méso-surface	95
5.2.3	La fonction de masquage d'une micro-surface décentrée	96
5.2.4	La BRDF d'une micro-surface décentrée	97
5.2.5	Validation du modèle décentré	98
5.3	Matériaux à nano-facettes	98
5.3.1	Le modèle de BRDF à nano-facettes	99
5.3.2	Le modèle de BRDF à nano-facettes pour les height maps	100
5.4	Résumé du chapitre	101

6	Etude de la corrélation azimuthale entre le masquage et l’ombrage dans le modèle de Smith	103
6.1	Introduction	104
6.2	Description de la configuration	105
6.2.1	Statistiques de la micro-surface	105
6.2.2	Notations probabilistes	105
6.2.3	La fonction de masquage	106
6.2.4	La fonction de masquage et d’ombrage	106
6.3	Dérivation de la fonction de masquage et d’ombrage locale	107
6.4	Dérivation de la fonction de masquage et d’ombrage distante	107
6.4.1	Dérivation de la fonction de masquage distante conditionnelle	108
6.4.2	Dérivation de la distribution conditionnelle des hauteurs	110
6.4.3	Propriétés de la corrélation azimuthale dans la fonction de masquage et d’ombrage distante	111
6.4.4	Introduction du coefficient de corrélation r_0	112
6.4.5	Approximation pratique de r_0	112
6.5	Expression finale du modèle proposé	116
6.6	Résultats	117
6.7	Résumé du chapitre	119
II	Représenter et évaluer les BRDFs multi-échelles	121
7	Cahier des charges pour une application de rendu multi-échelles	123
7.1	La contrainte de simplicité	123
7.2	Contraintes mémoire	124
7.3	Contraintes pour la représentation	125
7.4	Contraintes pour le calcul	126
8	Rappels sur les représentations des surfaces multi-échelles	127
8.1	Rappels généraux sur les représentations multi-échelles	128
8.1.1	Les représentations mono-lobe	128
8.1.2	Les représentations multi-lobes	128
8.1.3	Les représentations à lobes anisotropes	129
8.1.4	Les bases de fonctions	129
8.1.5	Les approches par clustering dépendant du point de vue	130
8.1.6	LEAN Mapping	130
8.2	Rappels et motivation pour les distributions de pentes à statistiques gaussiennes	131
8.2.1	Motivation, choix et limitations	131
8.2.2	Définition de la représentation	132
8.3	Compatibilité de la représentation avec l’animation	133
8.4	Résumé du chapitre	135
9	Représentation et évaluation du matériau multi-échelles	137
9.1	Le matériau dans les modèles à micro-facettes	137
9.1.1	Micro-surface à matériau constant	137
9.2	Représentation du matériau multi-échelles	138
9.2.1	Représentation du matériaux de base	138
9.2.2	Se ramener à une représentation linéairement combinable et séparable	139
9.3	Filtrage des color maps appliquées sur des textures	140
9.3.1	Rappels sur le filtrage des textures procédurales	141
9.3.2	Dérivation exacte de l’équation du filtrage des color maps	143

9.3.3	Représentation et calcul	144
9.3.4	Résultats	145
9.4	Filtrage des color maps appliquées sur des surfaces	146
9.4.1	Idées générales	146
9.4.2	Filtrage des color maps appliquées sur les hauteurs	149
9.4.3	Filtrage des color maps appliquées sur les normales	154
9.4.4	Exprimer l'intégration de l'éclairage comme un problème de filtrage de color map	158
9.5	A propos des approximations gaussiennes	160
9.6	Résumé du chapitre	163
10	Evaluation de la BRDF et de l'éclairage en temps-réel	165
10.1	Les équations d'illumination locale à évaluer	165
10.1.1	Modèles de BRDFs	165
10.1.2	Modèles d'éclairages	166
10.2	Evaluer les équations par intégration numérique	167
10.2.1	Discretisation des intégrales	167
10.2.2	Intégration de la BRDF à micro-facettes diffuses	169
10.2.3	Intégration de la BRDF à micro-facettes spéculaires	171
10.3	Résultats	174
10.4	Au-delà du modèle : attaquer les problèmes liés au filtrage de la géométrie	178
10.4.1	Unifier le filtrage de la courbure avec celui de la height map	178
10.4.2	Résoudre les problèmes liés à la tessellation adaptative	180
10.5	Résumé du chapitre	181
11	Importance sampling pour les BSDFs à micro-facettes	183
11.1	Le modèle de BSDF à micro-facettes	185
11.2	La technique précédente d'importance sampling	187
11.3	Notre technique d'importance sampling	189
11.4	Simuler $D_{\omega_i}(\omega_m)$ avec le modèle à V-cavités	190
11.4.1	Algorithme	190
11.4.2	Preuve de l'algorithme	191
11.5	Simuler $D_{\omega_i}(\omega_m)$ avec le modèle de Smith	192
11.5.1	Formulation dans l'espace des pentes	192
11.5.2	Invariance d'étirement de la distribution de pentes visibles	193
11.5.3	Simuler $P_{\omega_i}^{22}(x_{\bar{m}}, y_{\bar{m}}, 1, 1)$	196
11.5.4	Simuler $D_{\omega_i}(\omega_m)$	197
11.5.5	Simuler $D_{\omega_i}(\omega_m)$ avec le modèle de Smith décentré	199
11.6	Résultats et discussion	201
11.6.1	Validation	201
11.6.2	Analyse de convergence	201
11.6.3	Les cas très difficiles avec la méthode précédente	201
11.6.4	Les performances	202
11.7	Résumé du chapitre	208
III	Conclusion	209
12	Conclusion	211
12.1	Résumé des contributions	211
12.2	Limites et travaux futurs à court terme	211
12.3	Perspectives à long terme	213
12.4	Le Modèle Ultime de filtrage	214

IV	Annexes	221
A	Dérivation de la fonction de masquage et d’ombrage de Smith	223
A.1	Dérivation de la fonction de masquage	223
A.2	Dérivation de la fonction de masquage et d’ombrage corrélée par les hauteurs	226
A.3	Code MATLAB pour le Weak White Furnace Test	228
B	Implémentation MATLAB du facteur r_0 (chapitre 6)	233
C	Dérivations des jacobiens	235
C.1	Jacobian of the Normal-to-Slope Transformation	235
C.2	Beckmann Distribution	236
C.3	Jacobian of the Reflection Operation	237
D	Résultats supplémentaires pour l’intégration en temps-réel (chapitre 10)	239
D.1	Normal Mapping vs. Displacement Mapping	239
D.2	Our Shading Models and Comparisons with Ground Truth	241
E	Calcul des moments de la distribution de pentes visibles (chapitre 9)	249
E.1	Gaussian Surface Slope Distribution	249
E.2	Visible Slope Distribution	250
E.3	Solving for Integrals in Equations (E.30) to (E.38)	252
E.4	Implementation and Pseudocode	255
F	Résultats supplémentaires pour l’importance sampling (chapitre 11)	257
F.1	Generic Sampling Algorithm	257
F.1.1	C++ Implementation	258
F.2	Sparing Random Numbers	260
F.3	Sampling Beckmann Distributions	261
F.3.1	The Beckmann Distribution of Slopes	261
F.3.2	Simulate $P_{\omega_i}^{2-}(x_{\tilde{m}})$	261
F.3.3	Simulate $P^{2 2}(y_{\tilde{m}} x_{\tilde{m}})$	265
F.3.4	Simulate $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$ for Beckmann Distributions	266
F.3.5	C++ Implementation	267
F.3.6	Results	269
F.4	Sampling GGX Distributions	270
F.4.1	The GGX Distribution of Slopes	270
F.4.2	Simulate $P_{\omega_i}^{2-}(x_{\tilde{m}})$	270
F.4.3	Simulate $P^{2 2}(y_{\tilde{m}} x_{\tilde{m}})$	272
F.4.4	Simulate $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$ for GGX Distributions	274
F.4.5	C++ Implementation	275
F.4.6	Results	276
F.5	Convergence Comparisons	277
F.6	Results	280
F.6.1	Isotropic Beckmann	280
F.6.2	Anisotropic Beckmann	282
F.6.3	Isotropic GGX	283
F.6.4	Anisotropic GGX	285

Chapitre 1

Introduction

1.1 La complexité débordante des scènes en synthèse d’images

1.1.1 La synthèse d’images

La synthèse d’images est le domaine de la création d’images par ordinateur. Elle intervient dans les jeux vidéos, les effets visuels des films, la visualisation scientifique, etc. Créer une image par ordinateur nécessite plusieurs étapes dont l’étude constitue différents domaines de recherche.

Le domaine de la *modélisation* étudie comment créer l’ensemble du contenu des scènes 3D virtuelles. Il y a le contenu *géométrique* qui constitue les personnages de jeux vidéos, le décor virtuel d’un film ou encore une turbine pour un simulateur d’ingénierie mécanique, la configuration de l’*éclairage* avec différents types d’émetteurs de lumière et enfin les *modèles de matériaux* qui décrivent l’interaction entre la lumière et la géométrie.

Le domaine de l’*animation* est dédié à l’étude du mouvement du contenu des scènes 3D : le mouvement des personnages, d’une surface d’eau, de fumée ou de n’importe quel objet en mouvement dans une scène virtuelle.

Enfin, le domaine du *rendu*, auquel nous nous intéressons dans cette thèse, a pour objet la création de l’image elle-même. En particulier, le rendu photo-réaliste simule l’acquisition faite par un dispositif de capture d’images, comme un appareil photo, dans la scène virtuelle. Les algorithmes de rendu historiques étaient essentiellement basés sur des modèles empiriques qui permettaient de créer des images plausibles. Aujourd’hui, pour obtenir des images photo-réalistes qui ressemblent à de vraies photographies, la plupart des industriels utilisent des modèles de rendu *physically-based*, c’est-à-dire basés sur une modélisation physique. Concrètement, le rendu photo-réaliste consiste à faire une simulation de la physique de la lumière, aujourd’hui essentiellement basée sur des modèles de géométrie optique, dans la scène virtuelle. De fait, le rendu photo-réaliste est à l’intersection de la physique (modélisation quantitative de l’interaction entre la lumière et les matériaux), des mathématiques (résolution numérique des équations) et de l’informatique (implémentation d’algorithmes de calcul sur ordinateur). Le rendu photo-réaliste a énormément impacté l’industrie du cinéma. A titre d’exemple, cette année (2014), les auteurs du livre *Physically Based Rendering : From Theory to Implementation* [PH10] ont reçu un *Scientific and Technical Academy Award* pour son influence sur le milieu de la production de films.

1.1.2 Le problème du coût de la complexité

Dans le cadre de cette thèse, nous nous intéressons à la modélisation et au rendu des scènes *complexes*. Cette complexité se présente sous plusieurs formes, qu’on peut observer dans la figure 1.1 extraite du film Avatar. On trouve d’abord la complexité par la quantité de données utilisées pour générer l’image : la scène est immense, elle fait parfois plusieurs kilomètres carrés tout en étant très détaillée. Les détails sont modélisés au centimètre pour le décor, par exemple les feuilles des arbres existent individuellement, et en-dessous du



FIGURE 1.1 – Image extraite du film *Avatar* (Weta Digital ©).

millimètre pour les personnages pour lesquels les artistes créent jusqu’aux plus petites rainures de la peau. En plus de leur quantité, la nature et la spécification des données sont aussi complexes : il y a du contenu géométrique avec des modèles de matériaux sophistiqués mais aussi des volumes (la fumée, des nuages et même l’atmosphère) qui interagissent encore différemment avec la lumière. Les conditions d’éclairage sont aussi de plusieurs natures : sources directionnelles, éclairage environnemental distant, etc.

Le calcul du rendu dans une scène complexe de ce genre est un problème quantitativement difficile. Dans la forêt d’Avatar, il y a des milliers d’arbres contenant eux-mêmes des feuilles qui se projettent par milliers dans le même pixel. Un algorithme typique de rendu calcule la couleur de chaque feuille avant de moyenniser leurs contributions pour obtenir la couleur du pixel. Le calcul du rendu est d’autant plus fastidieux que la scène est complexe. Peut-on compter sur la puissance des ordinateurs pour gérer le calcul du rendu dans une scène d’une telle complexité ?

La célèbre loi de Moore est une conjecture, vérifiée depuis 50 ans, concernant l’évolution exponentielle de la puissance de calcul des ordinateurs. La prédiction initiale de Moore, exprimée en 1965, voulait que le nombre de transistors utilisés dans un ordinateur doublerait environ tous les dix-huit mois. Nous sommes effectivement habitués à disposer d’une puissance de calcul en augmentation constante, que ce soit dans nos ordinateurs, dans nos smartphones ou encore dans nos machines à laver. Pourtant, on se demande souvent, parfois avec une pointe d’ironie, pourquoi depuis vingt ans il faut toujours attendre cinq minutes entre le moment où on lance le logiciel de traitement de texte et le moment où on peut effectivement commencer à l’utiliser. Où est passée la puissance de calcul ? Si notre capacité de calcul augmente sans cesse, il en va de même pour l’utilisation que nous en faisons. Si le logiciel de traitement de texte met toujours cinq minutes à se lancer même sur une machine dix fois plus puissante, c’est, pour le dire simplement, parce qu’il fait des choses dix fois plus *complexes* : les fichiers sont plus gros et mettent plus de temps à charger, le nombre de polices de caractères et des diverses possibilités de mise en page est plus important, la correction orthographique est plus précise, les systèmes de révision ou de commentaires n’existaient pas auparavant, etc. La complexité de nos besoins informatiques augmente elle aussi de manière exponentielle. L’image de synthèse ne fait pas exception à la règle. James Blinn, un des pionniers du domaine, prédisait que le temps passé à calculer une image serait toujours le même, peu importe la puissance de calcul des ordinateurs. Cette prédiction est connue sous le nom de la loi de Blinn :

”As technology advances, rendering time remains constant.”

Blinn’s Law

En effet, les artistes et les ingénieurs qui créent les images sont habitués à avoir un certain budget de temps de calcul par image. Un ordinateur plus puissant permet de gagner du temps immédiatement réinvesti dans la complexité de l'image : les artistes créent des modèles plus détaillés dans des scènes plus grandes, les ingénieurs calculent des simulations plus précises, etc. L'équilibre entre la puissance des ordinateurs et la complexité prédit par la loi de Blinn fut constaté pendant quinze ans chez Pixar, où le temps de calcul moyen d'une image fut d'environ dix heures. Cependant, cet équilibre n'est plus tout à fait d'actualité.

Dans le monde de la production, les ambitions aujourd'hui sont telles que l'augmentation de la complexité a déjà dépassé celle de la puissance de calcul des ordinateurs. Les artistes ne créent plus de simples scènes mais des mondes immenses et détaillés qui deviennent presque impossibles à rendre dans le budget de temps habituel, même avec des machines dernier cri. La complexité des données est telle que même les opérations les plus élémentaires, comme les charger ou les sauvegarder avant même de les utiliser pour des calculs, sont devenues complexes.

Dans le monde du temps-réel, on se heurte de plus en plus aux limites physiques du hardware. La problématique de l'énergie est aujourd'hui d'une importance égale, voire supérieure, à celle des performances pour les constructeurs. Pour garder un temps de rendu constant, la question d'actualité n'est plus de savoir comment faire plus avec plus, comme le voulait la loi de Blinn, mais plutôt de savoir comment faire plus avec moins.

Le constat général est qu'en image de synthèse la complexité des scènes augmente plus vite que la puissance brute des ordinateurs : la loi de Blinn ne tient plus. En conséquence, un défi important pour le futur de l'image de synthèse, et de l'informatique en général, est d'évoluer vers de nouvelles manières de gérer cette complexité.

1.1.3 Objectif de la thèse

Dans le cadre de cette thèse, nous nous intéressons au rendu des scènes tellement *détaillées*, c'est-à-dire avec des détails en si grand nombre (complexité des données) et interagissant de manière si complexe avec la lumière (complexité des calculs), qu'elles ne pourraient pas être rendues en temps raisonnable en comptant simplement sur la puissance des machines.

Notre objectif général est de créer des algorithmes et des représentations qui permettent une gestion plus efficace des détails, dans le but de rendre possible le calcul d'images à partir de scènes très détaillées en temps raisonnable tout en garantissant qu'elles soient de qualité.

L'idée générale qui est soutenue dans cette thèse est que, bien que la quantité d'information brute soit immense dans les scènes très détaillées, la quantité d'information contribuant effectivement à la valeur d'un pixel est bien moins importante. Alors que la plupart des algorithmes existants font des calculs qui impliquent toutes les données de la scène en même temps, nous pensons qu'en trouvant la bonne manière de représenter et de calculer l'information utile à la couleur des pixels, il est possible de créer des algorithmes beaucoup plus efficaces et précis.

Pour reprendre l'exemple de la forêt dans Avatar, alors que l'ordinateur ferait un calcul impliquant toutes les feuilles de tous les arbres pour calculer l'apparence d'une forêt lointaine, un peintre s'y prendrait d'une autre manière. Il imaginerait plutôt quel est l'aspect du matériau "forêt" et la peindrait directement avec le bon niveau de détails visibles dans l'image. En effet, le peintre utilise intuitivement la notion d'*échelle* : il choisit le niveau de détail directement observable dans l'image. Au contraire, l'ordinateur ne connaît en général que l'échelle *micro*, c'est-à-dire l'échelle la plus fine, celle des données, et n'a pas accès aux échelles supérieures. Si, comme le peintre, l'ordinateur avait accès à l'apparence de la forêt à l'échelle désirée, il pourrait calculer directement les valeurs des pixels sans avoir à faire un calcul exhaustif pour toutes les feuilles. Sur ordinateur, c'est le pixel qui donne la notion d'échelle. Idéalement, nous souhaitons mettre au point un modèle de représentation et de calcul qui soit capable de décrire à la fois les détails de manière individuelle lorsqu'on les voit de près (la feuille) mais aussi de manière groupée lorsqu'on les voit de loin (le feuillage). De cette manière, les calculs ne seraient faits qu'à l'échelle utile, donnée par la taille du pixel, et la complexité serait maîtrisée. C'est ce qu'on appelle un modèle de calcul *multi-échelles*.

Si les modèles multi-échelles semblent être une solution à la gestion de la complexité, ils sont très peu utilisés en pratique dans les moteurs de rendu. La raison est simple : concevoir des modèles multi-échelles

est difficile. En effet, les équations de base du rendu qui décrivent les interactions à l'échelle micro (la feuille) sont aujourd'hui bien maîtrisées dans le monde industriel qui est très proche de l'état de l'art académique. En revanche, les échelles supérieures (le feuillage, la forêt) sont souvent plus difficiles à modéliser. Mettre au point un modèle physique multi-échelles et y associer une représentation et un modèle de calcul efficace est un exercice compliqué qui nécessite une compréhension pointue à la fois de la physique, des méthodes de calcul et des architectures matérielles des ordinateurs. Aujourd'hui, le modèle multi-échelles "ultime", qui serait capable de prendre en compte tous les éléments d'une scène en même temps, est hors de portée. Pour rendre compte du tout, il faut commencer par maîtriser les parties, dont beaucoup de briques intermédiaires sont encore manquantes. En conséquence, aujourd'hui, il est plus raisonnable de commencer par mettre au point des modèles multi-échelles spécifiques à des phénomènes particuliers (la forêt, par exemple), tout en gardant à l'esprit que ce ne sont que de petits pas vers le Modèle Ultime.

Cette thèse est dédiée à l'étude des modèles multi-échelles pour le cas particulier des surfaces détaillées, comme la peau de personnages, la surface de la mer ou des objets. C'est un problème intermédiaire riche et intéressant, peu exploré dans la littérature, et qui est en même temps un enjeu en soi pour l'industrie de l'image de synthèse qui utilise massivement ces représentations surfaciques en production. La figure 1.2 est un exemple de résultat obtenu avec les techniques développées dans cette thèse et appliquées sur un modèle de production de Disney.

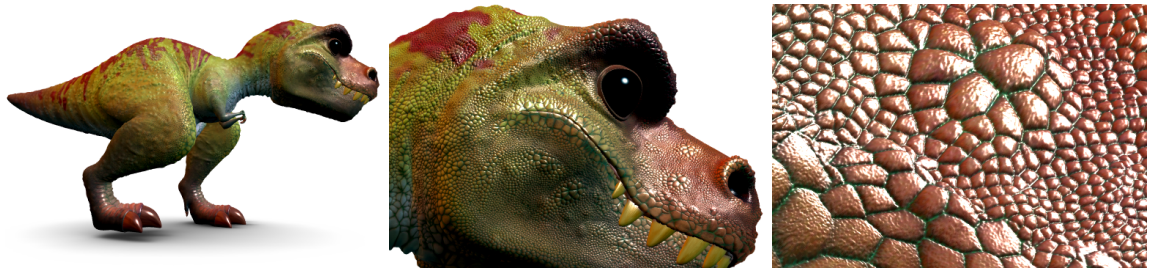


FIGURE 1.2 – Rendu multi-échelles de surface détaillée [DHI⁺ 13]. Pour maîtriser la complexité des détails, l'algorithme de rendu adapte leur échelle en fonction de la distance à l'écran et effectue toujours le même calcul par pixel.

1.2 Approche générale

Dans la communauté du rendu, une critique récurrente faite aux modèles multi-échelles est qu'ils sont intrinsèquement mal posés et produisent des résultats dont la qualité est forcément mauvaise. Le but de cette section est de montrer que cette affirmation est fautive et vient d'un amalgame entre le modèle mathématique du rendu multi-échelles et ses implémentations existantes qui font certaines approximations. Pour ce faire, nous abordons le problème du rendu comme un problème générique d'intégration numérique en nous affranchissant momentanément des représentations et des implémentations existantes en image de synthèse. De cette manière, nous spécifions formellement notre approche et montrons qu'elle est mathématiquement bien définie. C'est cette définition formelle correcte qui servira par la suite de point de départ pour mettre en oeuvre des modèles utilisables en pratique sans faire d'hypothèses clandestines et éviter certains pièges classiques.

1.2.1 Le rendu est un problème d'intégration

La plupart des équations du rendu sont exprimées sous forme intégrale et n'ont pas de solution analytique. Par exemple, les équations d'illumination locale et du rendu volumique sont illustrées dans la figure 1.3.

Formellement, en rendu, on peut voir la spécification de la scène (le contenu, l'éclairage et les matériaux) comme les paramètres d'une fonction générique f , les valeurs des pixels comme l'intégrale de f sur le support

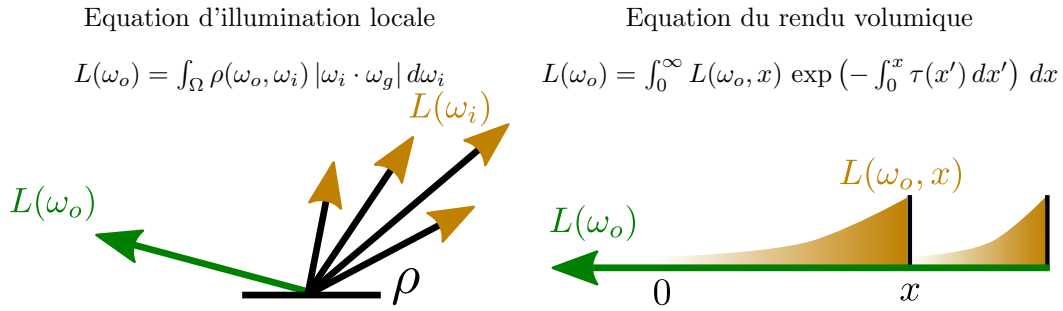


FIGURE 1.3 – Exemples d'équations du rendu sous forme intégrale.

du pixel (possiblement modulé par un certain filtre), et les algorithmes de rendu comme des algorithmes d'intégration numérique. Cette idée est illustrée en figure 1.4. C'est sous cet angle que nous analysons les propriétés des algorithmes d'intégration numérique dans la suite de cette section.

$$\int_{\text{pixel}} \overset{f(\text{geometry, material, lighting, etc.})}{\text{wavy line}} = \underset{\text{pixel value}}{?}$$

FIGURE 1.4 – La radiance d'un pixel est l'intégrale d'une fonction qui dépend du contenu de la scène.

1.2.2 La limite des approches “brute force”

Les algorithmes d'intégration “brute force” consistent à utiliser un nombre d'échantillons suffisant pour capturer toute la complexité de la fonction à intégrer : c'est le contenu de la scène qui asservit la complexité de l'intégration, et donc le temps de calcul. Ils garantissent un résultat exact (ou du moins avec une erreur qui tend vers 0) mais dont le temps de calcul peut être très élevé.

Il existe plusieurs manières de formaliser l'asservissement de l'intégration par la complexité des données. Une manière de voir les choses est de regarder le problème dans le domaine de Fourier : le théorème de Nyquist-Shannon nous dit que pour intégrer ou reconstruire correctement un signal, il faut l'échantillonner à une fréquence qui dépend de la fréquence maximale du signal¹. Cette manière d'intégrer est illustrée en figure 1.5. Si la condition d'échantillonnage n'est pas respectée, la reconstruction est erronée : c'est le problème de l'*aliasing*. C'est à travers cette condition sur la fréquence minimale d'échantillonnage que le contenu de la scène asservit l'intégration. Une autre manière de voir les choses est l'approche stochastique : les données sont vues comme un signal aléatoire dont on estime la moyenne avec des techniques d'intégration stochastiques de type Monte Carlo. Le nombre d'échantillons à utiliser dépend de la variance du signal. C'est à travers leur variance que les données asservissent l'intégration.

Les approches “brute force” sont souvent utilisées dans le monde de la production pour les films où la qualité d'image est plus importante que le temps de calcul. Ces approches sont viables lorsque la loi de Blinn peut être tenue, c'est-à-dire lorsque la puissance de l'ordinateur est assez importante pour traiter la complexité des données en temps raisonnable. Lorsque cette hypothèse n'est plus valide, quand la complexité des données surpasse complètement la puissance de l'ordinateur, ces approches ne sont plus utilisables, le temps de calcul nécessaire pour atteindre la convergence étant trop important.

1. Il faut échantillonner le signal à au moins deux fois sa fréquence maximale pour le *reconstruire* et, nuance, à au moins une fois sa fréquence maximale pour l'*intégrer*.

$$\int_{\text{pixel}} f(\text{geometry, material, lighting, etc.}) = \left\langle \frac{\text{signal}}{\text{pixel}}, \text{sample grid} \right\rangle = \sum \text{sampled signal}$$

FIGURE 1.5 – Le théorème d’échantillonnage de Nyquist-Shannon : l’échantillonnage doit être assez fin pour capturer toutes les fréquences du signal.

Définition : une scène complexe est une scène qui ne peut pas être rendue en temps raisonnable par une approche “brute force”, en comptant uniquement sur la puissance de calcul de la machine.

Les méthodes adaptatives. Certaines techniques d’intégration sont adaptatives, ce qui signifie que l’asservissement de l’intégration par les données n’est pas ajusté de la même manière partout : il est ajusté adaptativement en exploitant le fait que certaines parties de la scène peuvent être moins complexes que d’autres. D’un point de vue spectral, on pourrait dire que ces zones contiennent moins de hautes fréquences, ou encore, d’un point de vue stochastique, on pourrait dire qu’elles ont une variance moins importante. En conséquence, l’intégration demande moins de calculs dans ces parties moins complexes. En adaptant les calculs à chaque partie de la scène, les algorithmes adaptatifs augmentent les performances tout en restant dans le critère de qualité.

A titre d’exemple, j’ai travaillé sur une méthode adaptative pendant mon stage de M2 [SHUD10] : pour accélérer l’intégration de rendus volumiques, nous utilisons un critère adaptatif pour choisir le pas d’intégration numérique. Le critère d’adaptativité était calculé justement à partir du contenu fréquentiel des données.

La limitation de ces approches est contenue dans leur postulat de base : certaines parties de la scène sont peu complexes pour un certain critère, par exemple elles sont basses-fréquences ou encore à faible variance. C’est sous cette hypothèse seulement qu’un gain est possible. Pour les scènes uniformément complexes, les approches adaptatives ne sont pas plus efficaces que les approches non adaptatives, voire pire, la gestion de l’adaptativité entraînant parfois un surcoût, autant en terme de complexité algorithmique (opérations supplémentaires), que de mémoire (structures de données supplémentaires), ou encore de développement (bugs supplémentaires). Dans le cas général, le gain est un facteur qui dépend de la nature de la scène, mais qui ne rend pas pour autant l’algorithme indépendant de la complexité de la scène : une scène deux fois plus complexes exige deux fois plus de calculs. En conséquence, ces méthodes ne constituent pas une solution pour les scènes dont la complexité dépasse la puissance de calcul disponible.

1.2.3 Le modèle du rendu filtré

Pour maîtriser la complexité des données, nous souhaitons mettre au point des algorithmes *scalables*. Attention, en *High Performance Computing* (HPC) le mot “scalabilité” et l’expression “passer à l’échelle” signifient souvent que la complexité d’un algorithme augmente de manière linéaire avec celle des données : un problème deux fois plus complexe demande deux fois plus de ressources. Dans le domaine du rendu, nous donnons au mot “scalabilité” une définition différente.

Définition : en rendu, un algorithme de rendu scalable est un algorithme dont la complexité dépend du nombre de pixels dans l’image à calculer et non des données : la fréquence d’échantillonnage est constante et proportionnelle au nombre de pixels.

Cette nuance de définition fait toute la différence lorsque la loi de Blinn ne tient plus : les algorithmes scalables au sens du HPC ne permettent pas de résoudre le problème de la complexité quand les données augmentent plus vite que la puissance de calcul disponible. En conséquence, il est donc primordial que la

complexité de l'algorithme ne dépende pas de celles des données mais plutôt d'une métrique d'apparence, qui est celle de l'image.

A échantillonnage constant, seule solution pour remplir la condition de scalabilité, on joue sur l'autre paramètre de la condition de Shannon-Nyquist : pour éviter l'aliasing, c'est à la scène d'adapter son contenu fréquentiel à une fréquence d'échantillonnage spécifique. Pour sous-échantillonner un signal f sans introduire d'aliasing, il faut d'abord lui appliquer un filtre passe-bas k qui supprime les fréquences supérieures à la fréquences d'échantillonnage :

$$\int_{-\infty}^{+\infty} f(x) dx = \sum_i (f * k_{\Delta x})(i \Delta x).$$

L'opération de *filtrage* s'exprime mathématiquement par la convolution du signal f par un filtre $k_{\Delta x}$. En pratique, pour le rendu, les échantillons $i\Delta x$ sont les pixels de l'écran, dont le filtre de reconstruction est $k_{\Delta x}$, et l'intégrale de la fonction f à calculer est la radiance mesurée par les pixels. Le principe du rendu filtré est de n'évaluer qu'une seule fois la fonction $f * k_{\Delta x}$ par pixel, ce qui garantit un rendu exact et scalable : la complexité de l'algorithme dépend du nombre d'évaluation de $f * k_{\Delta x}$, c'est-à-dire uniquement du nombre de pixels et pas du contenu f de la scène.

Cette équation n'est pas une approximation, c'est une équation exacte, si le filtre $k_{\Delta x}$ est choisi correctement en fonction de la période d'échantillonnage Δx . S'il est possible d'évaluer la fonction filtrée $f * k_{\Delta x}$, alors il est possible d'intégrer exactement f avec un nombre arbitraire d'échantillons x_i , la difficulté principale étant l'évaluation efficace de $f * k_{\Delta x}$.

The diagram shows the mathematical principle of filtered rendering. On the left, an integral over a pixel of a signal f (representing geometry, material, lighting, etc.) is shown. This is equated to a convolution operation: the signal f is convolved with a kernel k (represented by a bell curve), and the result is sampled at discrete points (represented by red arrows). The final result is a sum of the filtered signal at those points.

FIGURE 1.6 – Le principe du rendu filtré est d'appliquer un filtre passe-bas avant d'échantillonner le signal.

Définition. Le rendu filtré consiste à adapter le contenu fréquentiel de la scène à la fréquence d'échantillonnage fixée à l'avance : c'est la résolution requise pour l'image qui asservit la complexité de la scène. Cette approche rend l'algorithme scalable.

Qualité vs. performances : un faux dilemme. Le choix entre un algorithme brute force ou avec filtrage est souvent présenté sous la forme d'un compromis : souhaite-t-on avoir plutôt des images de qualité ou avoir de bonnes performances ? Pourtant, nous avons vu que, formellement, les deux approches garantissent un résultat exact, et ainsi, la même qualité d'image. Le malentendu vient d'un amalgame entre le modèle du rendu filtré et les implémentations qui en sont faites. La clé du rendu filtré est de trouver un moyen de définir et d'évaluer efficacement la fonction filtrée $f * k_{\Delta x}$, ce qui est un problème difficile dans le cas général. Pour simplifier, diverses approximations sont faites en pratique. Par exemple, pour les textures, l'approche standard est d'utiliser une pyramide de MIPmap [Wil83], où chaque niveau possède un contenu fréquentiel deux fois moins important que celui du niveau précédent. La géométrie, quant à elle, est souvent présente en plusieurs exemplaires avec des niveaux de simplification et des nombres de primitives variables. Mais il faut garder à l'esprit que ces méthodes *ne sont pas* le rendu filtré, elles ne sont que des implémentations possibles, souvent datées et anciennes, s'appuyant sur de fortes approximations. L'idée principale qui motive les travaux présentés dans cette thèse est la suivante :

Le compromis qualité/performance est un faux dilemme. Il est possible de créer des algorithmes de rendu filtré qui garantissent un rendu scalable et haute-qualité.

1.2.4 Les fausses pathologies du rendu filtré

Avant de proposer une méthodologie générale pour mettre au point des algorithmes de rendu filtré, nous allons voir certaines pathologies des implémentations existantes et les pièges à éviter.

Pathologie 1 : Simplifier les données. Dans une scène typique d'un rendu temps-réel, une stratégie courante est de réduire le nombre de détails à afficher à l'écran : ce processus s'appelle la *simplification*. La stratégie de simplification rend l'algorithme scalable car elle asservit le nombre de primitives à afficher, mais elle ne garantit pas la qualité du rendu. En effet, le processus de simplification est destructif. D'un point de vue de traitement du signal, la simplification revient à supprimer de l'information du signal, comme illustré par la figure 1.7. Le calcul de l'intégrale, qui ignore les informations perdues, ne peut donc pas garantir de calculer la bonne valeur. Contrairement au rendu filtré, la simplification introduit une erreur, car elle ne traite pas correctement la quantité d'information présente dans la scène : *simplifier n'est pas filtrer*.

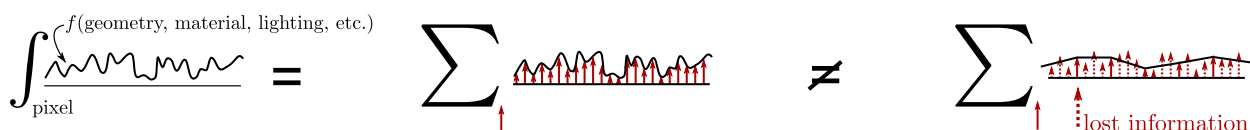


FIGURE 1.7 – La simplification supprime de l'information du signal.

Un exemple de simplification est donné en figure 1.8 : pour éviter l'aliasing, les petits détails sont simplifiés (supprimés) de la scène au fur et à mesure du zoom arrière. Cette suppression pure et simple crée des artefacts visibles au moment de la transition.

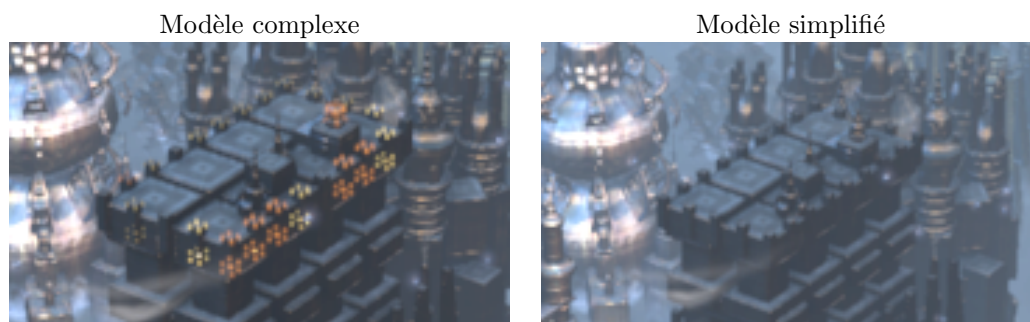


FIGURE 1.8 – Exemple de pathologie 1 : Supprimer des données de la scène. Dans cet exemple, les lumières du bâtiment disparaissent subitement lorsque le niveau de détail change (Démonstration "Endless City" de NVIDIA ©).

Pathologie 2 : filtrer linéairement les données avant un traitement non linéaire. L'apparence d'un modèle 3D est rarement celle des données brutes mais plutôt le résultat de calculs complexes qui prennent les données comme input : la fonction à filtrer n'est pas les données, mais la fonction obtenue après l'application de tous les traitements non linéaires sur les données. En général, ces traitements sont des fonctions non linéaires appliquées sur les données. A cause de cette non linéarité, le traitement ne commute pas avec l'opération de filtrage dans le cas général, ce qui s'écrit mathématiquement :

$$\int f(g(x)) dx \neq f\left(\int g(x) dx\right).$$

La figure 1.9 montre un exemple où les données représentent une fonction de bruit qu'on sait filtrer correctement. Mais les fonctions de bruits sont rarement utilisées seules pour créer des textures procédurales qui subissent souvent des traitements non linéaires. Un exemple de traitement non linéaire est l'application d'une

color map sur la fonction de bruit. Appliquer la color map sur le bruit filtré linéairement ne produit pas la bonne apparence, car ce n'est pas équivalent à filtrer la texture procédurale après application de la color map. Un autre exemple du même type est donné en figure 1.10 : une normal map est utilisée pour enrichir le contenu de la scène en détails visuels. Calculer le shading sur la normal map filtrée linéairement ne produit pas la même apparence globale qu'avec la texture haute résolution : des reflets spéculaires ont disparu et l'aspect général de l'image de droite est plus sombre que celle de gauche. C'est parce que le shading est une opération non linéaire appliquée sur les normales qui ne commute pas avec le filtrage de la normal map.

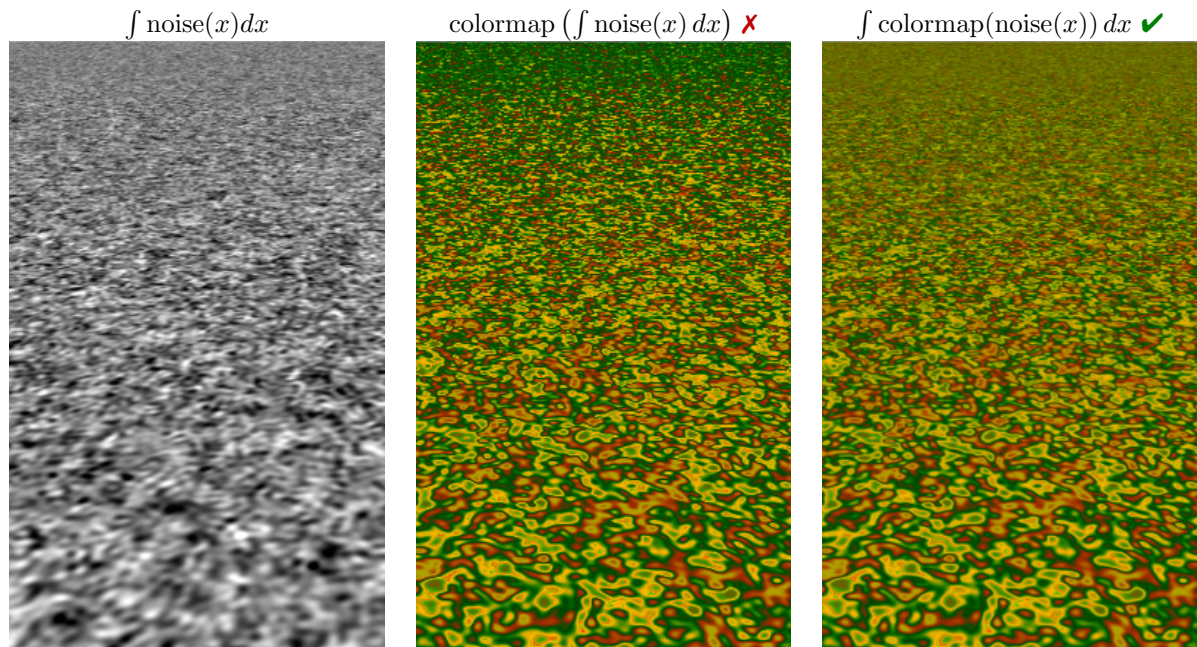


FIGURE 1.9 – Exemple de pathologie 2 : appliquer une color map non linéaire sur une fonction de bruit filtrée [HNPN13a].

Pathologie 3 : filtrer séparément des données non séparables. En plus d'être potentiellement transformées par des traitements non linéaires, les données peuvent aussi interagir avec d'autres données dans le calcul final de l'image. Lorsque les données sont corrélées, elle ne peuvent pas être filtrées séparément, ce qui mathématiquement s'écrit :

$$\int f(x) g(x) dx \neq \int f(x) dx \times \int g(x) dx.$$

Un exemple est proposé dans la figure 1.11 : la surface du dragon est faite de détails géométriques très fins et corrélés à la couleur. Les détails les plus visibles de la surface (les bosses), sont marron alors que les parties les moins visibles (les trous) sont vertes. Cette corrélation produit des effets visibles de loin dans les silhouettes. En effet, à cause des effets de masquage, seuls les détails les plus visibles (les bosses marron) contribuent à l'apparence des silhouettes. Filtrer la texture de couleur sans tenir compte de cette corrélation ne produit pas la même apparence globale, car la couleur moyenne de la texture n'est pas la couleur moyenne visible.

Partir des données est source de problème. Le point commun des trois pathologies que l'on vient de voir est d'aborder la question du filtrage en partant des données. Par exemple, la préface du livre *Level of Detail for 3D Graphics* présente le problème sous cet angle [LWC+02] :

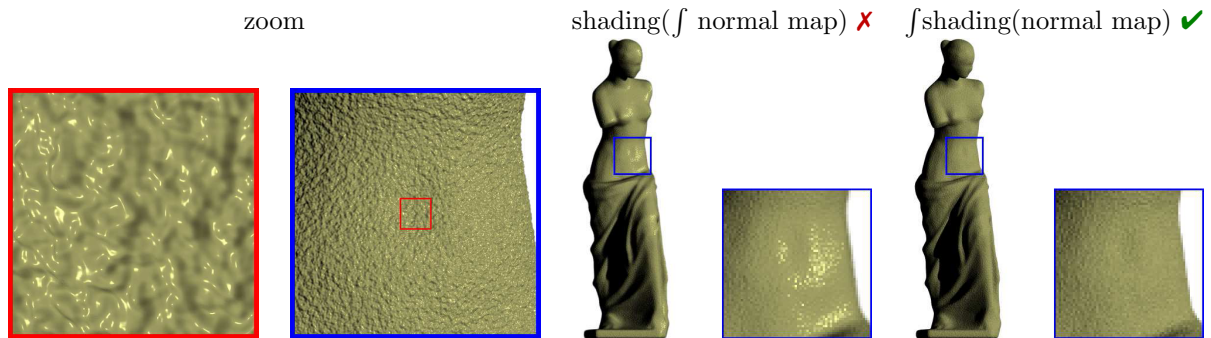


FIGURE 1.10 – Exemple de pathologie 2 : calculer le shading sur une normal map filtrée [HN12].

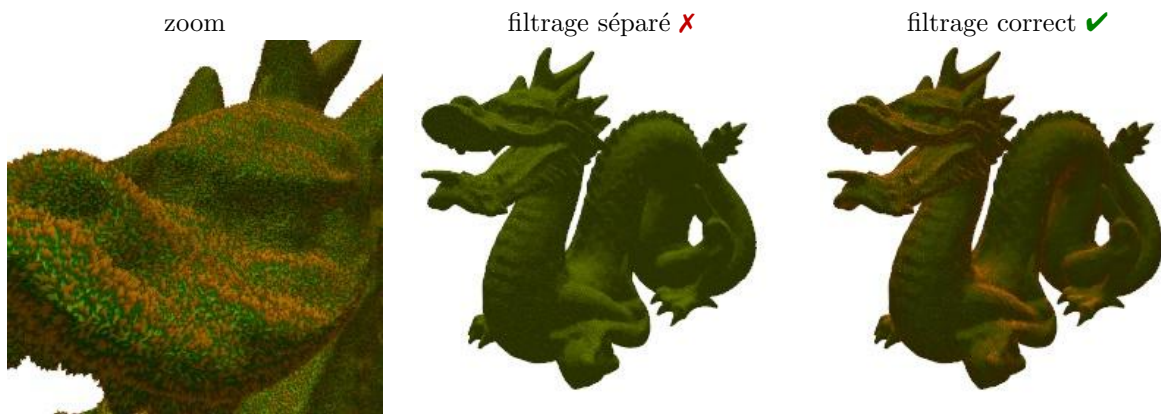


FIGURE 1.11 – Exemple de pathologie 3 : Filtrer la texture sans tenir compte des corrélations avec la visibilité [HNP13a]. Dans cet exemple, la texture est corrélée avec la hauteur des détails géométriques de la surface : les bosses sont brunes et les trous sont verts. Dans les silhouettes, seules les bosses, et donc la couleur brune, sont visibles. Cet effet disparaît lorsque la texture est filtrée sans tenir compte de la corrélation entre la couleur et la visibilité.

” The strategy by which the overall challenge has been tackled is conceptually very simple : People devise algorithms so that the primitives actually fed into the graphics pipeline are (almost) only those that will be seen and noticed by the viewer of the final image.

...

Many unobscured triangles in a field of view will nevertheless be invisible to the viewer. At their distance to the viewpoint, their projection on the final image will be substantially less than the pixel resolution of the display. So there is no need to burden the graphics pipeline with those.

Instead, one wants to manage the world model so that fine levels of detail are grouped together into a single primitive when, and only when, they would not be seen separately.”

Level of Detail for 3D Graphics

En résumé : les données étant trop nombreuses, il faut réduire leur nombre, en les simplifiant ou en les filtrant. Mais c’est en s’attaquant aux données directement sans tenir compte de leur contribution à l’image qu’on introduit des erreurs. En effet, si le processus de simplification ou de filtrage des données détruit de l’information pertinente pour l’image, calculer la bonne image devient un problème mal posé et la partie est

perdue d'avance.

1.2.5 Vers un modèle de rendu filtré bien posé

Le pixel comme seule métrique de qualité. L'approche générale que nous proposons consiste donc à toujours partir de l'image finale, ce qui en pratique veut dire partir de l'équation du rendu qui donne la valeur des pixels pour le modèle qu'on considère, que ce soit des surfaces, des volumes, ou n'importe quoi d'autre. Pour un modèle de rendu donné, notre démarche consiste à poser l'équation qui permet de calculer la couleur d'un pixel. L'équation est souvent compliquée et, en pratique, devra souvent être approximée. Néanmoins, c'est l'analyse de cette équation qui sert de point de départ et de métrique de qualité pour choisir des approximations raisonnables. De cette manière, chaque hypothèse d'approximation est explicitée et utilisée en connaissance de cause. Cette approche n'exclut pas l'analyse des données : lorsqu'elles présentent des propriétés particulières, nous pouvons bien sûr en tirer partie. Mais, une fois de plus, c'est le pixel seul qui sert de métrique pour décider si une propriété des données peut-être utilisée pour faire des approximations raisonnables ou non.

Trouver l'information pertinente. Pour faire des approximations raisonnables, il faut trouver les informations pertinentes pour l'image. Un postulat du rendu filtré est que la quantité d'information pertinente pour calculer l'image est beaucoup plus faible que celle présente dans la scène entière. Cette propriété est souvent vérifiée dans les scènes réelles où l'effet émergent des détails agrégés possède une dimensionalité plus faible que l'ensemble des détails. Par exemple, l'apparence de la forêt n'est pas un bruit blanc incompressible, elle obéit à un certain ordre et à certaines propriétés qui émergent de l'organisation des milliers de feuilles. C'est cette organisation globale qui est pertinente pour l'image et elle est généralement beaucoup plus compacte que l'énumération de l'ensemble.

Le principe du rendu filtré est de déterminer les modèles émergents qui décrivent cette organisation. Ils s'expriment souvent sous la forme d'une physique statistique des modèles sous-jacents. Une approche typique consiste à transformer l'équation exprimée dans l'espace des données en une équation statistique exprimée dans l'espace des propriétés des données.

Trouver le bon modèle émergent n'est pas facile dans le cas général. Plus on dispose d'informations a priori sur le modèle, plus il est facile de trouver celle qui est pertinente. C'est pour cette raison qu'en général, travailler sur des modèles très spécifiques est plus simple et permet d'aller plus loin dans l'efficacité du modèle. En effet, plus le modèle est spécifique (par exemple, il ne modélise que l'océan), plus on dispose de connaissances a priori à exploiter (on sait que l'océan a toujours la propriété X ou la propriété Y). Un modèle très spécifique permet donc l'exploitation de beaucoup de connaissances a priori, ce qui le rend efficace et précis. Au contraire, les modèles plus généraux laissent moins de connaissances a priori à exploiter. En pratique, l'information manquante est souvent remplacée par plus de calculs ou plus de mémoire, ce qui rend ces modèles moins efficaces.

Exemples de travaux précédents sur des cas très spécifiques. Eric Bruneton et Fabrice Neyret ont appliqué cette méthodologie à des cas très particuliers, comme le rendu de l'océan ou de forêts [BNH10, BN12b]. Les images de la figure 1.12 ont été créées avec le logiciel Proland d'Eric Bruneton, qui permet une exploration temps-réel de ce type de paysages. Dans le cas de l'océan animé, lorsque la caméra est suffisamment proche pour voir les vagues, elles sont rendues individuellement par le logiciel. Lorsqu'elles sont trop loin pour qu'on puisse les distinguer, l'algorithme ne les rend plus une par une, mais rend directement un ensemble de vagues, en se basant sur leur comportement optique statistique. Le modèle est capable de décrire à la fois la vague, mais aussi la physique statistique qui émerge d'un ensemble de vagues, grâce à une représentation unifiée qui permet une navigation continue entre ces différentes échelles. De cette manière, la complexité de l'océan est complètement maîtrisée par l'algorithme. Il en va de même pour le rendu de la forêt.

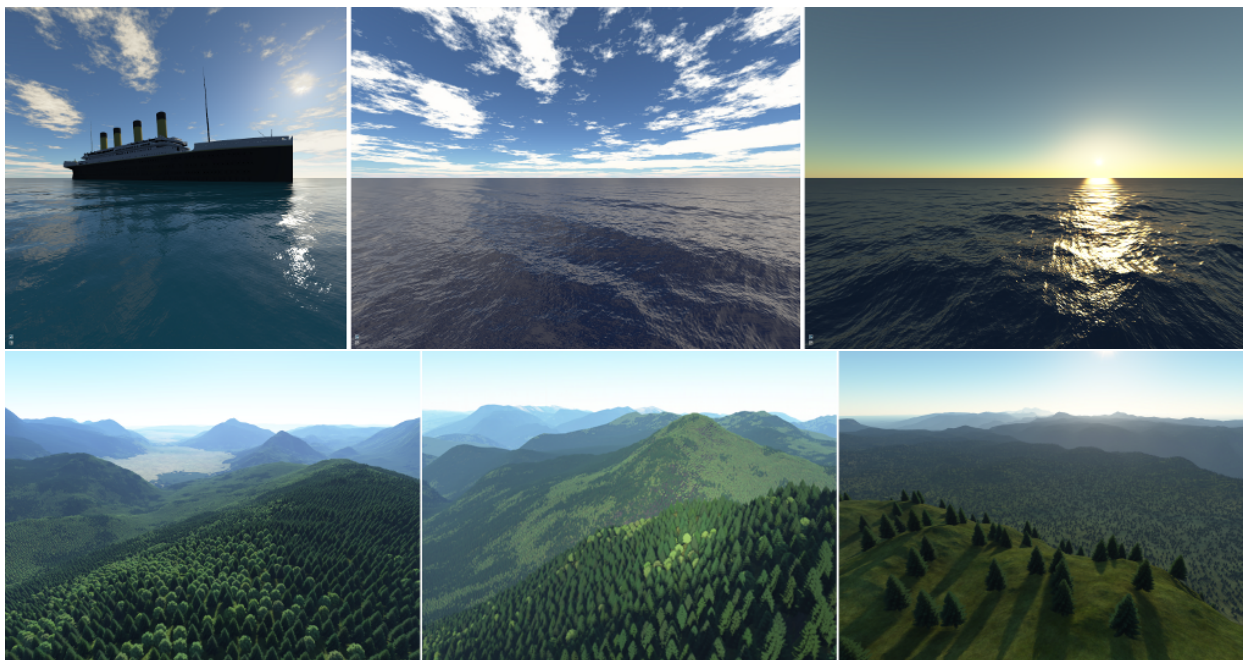


FIGURE 1.12 – Exemples de représentations filtrées pour le rendu temps-réel de qualité dans le logiciel Proland [BNH10, BN12b].

1.3 Périmètre thématique

Dans cette section, nous motivons le choix des détails surfaciques comme primitive de rendu multi-échelles.

1.3.1 Critères pour choisir un modèle d'étude

L'objet de cette thèse est de concevoir des modèles de rendu multi-échelles qui peuvent s'appliquer à une classe de scènes ou d'objets plus large que des cas aussi particuliers que la surface de l'océan. D'un autre côté, viser une classe de problèmes trop générale est peu prometteur, puisque les modèles multi-échelles sont efficaces lorsqu'on dispose de connaissances a priori à exploiter. Il s'agit donc de trouver un compromis intéressant entre trop spécifique et trop général. Nous souhaitons trouver un modèle qui satisfasse les critères suivants :

1. Assez ouvert pour être exploitable pour différents problèmes utiles à de nombreuses scènes d'image de synthèse dans le monde de la production.
2. Assez spécifique pour pouvoir exploiter des connaissances a priori suffisantes pour atteindre les objectifs de scalabilité et de qualité.
3. Qui permette de développer des connaissances et une méthodologie applicables à d'autres types de modèles : nous souhaitons que la démarche soit féconde pour des travaux futurs.

Il est assez naturel d'attaquer la question du filtrage là où on peut tirer partie d'une certaine organisation. Les scènes 3D sont généralement composées de surfaces, représentées par des triangles qui n'ont pas d'organisation a priori, ce qui rend la question du filtrage de l'apparence très complexe en général. En revanche, les détails les plus fins ne sont souvent pas représentés directement de manière géométrique, mais plutôt sous la forme d'une amplification de la géométrie existante. Cette amplification, pour pouvoir être construite puis appliquée correctement, est souvent contrainte, organisée, paramétrée d'une certaine manière qui apporte de

la connaissance a priori exploitable. Ces représentations amplificatrices aux petites échelles sont donc des parties de choix pour aborder la question du filtrage. Dans le cadre de cette thèse, nous nous intéressons au filtrage de l'apparence des détails surfaciques.

1.3.2 Les détails surfaciques comme primitives multi-échelles

Nous nous intéressons au cas des surfaces amplifiées par des détails stockés dans des *textures* appliquées sur la surface via une *paramétrisation locale*, typiquement des coordonnées de texture (u, v) . Les détails surfaciques sont associés à des échelles différentes : *macro-*, *méso-* et *microscopique*. Le principe du rendu multi-échelles est la navigation continue entre ces échelles en fonction de l’empreinte du pixel : c’est le pixel qui sert de métrique pour définir les différentes échelles.

L’échelle macroscopique : la géométrie. L’échelle macroscopique est celle à laquelle les détails géométriques sont observables et distinguables individuellement : c’est l’échelle sur-pixel. Dans notre cas, elle correspond à l’amplification de la géométrie via une *height map* : une texture haute résolution qui stocke des valeurs de hauteur utilisées pour déplacer la surface basse-fréquence de manière à lui ajouter des détails hautes-fréquences (toujours visibles, jamais sous-pixel), comme illustré dans la figure 1.13.

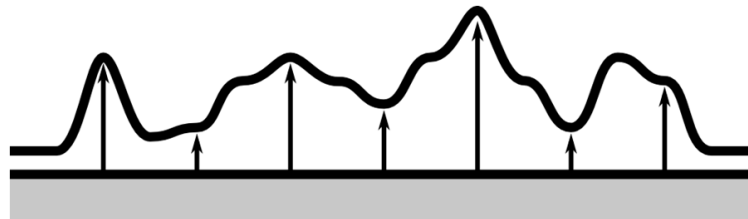


FIGURE 1.13 – Les *height maps* représentent une fonction de hauteur utilisée pour amplifier la géométrie.

Les *height maps* peuvent s’appliquer sur différents types de représentations surfaciques, que ce soit des maillages, des surfaces implicites, etc. Ce qui sera développé dans cette thèse ne fait pas de supposition sur le type de représentation sous-jacente, mais, typiquement, ce sont les maillages qui sont utilisés : le pipeline de rendu augmente la résolution du maillage de base avec un algorithme de tessellation et les nouveaux sommets sont déplacés orthogonalement à la surface avec les valeurs de hauteur de la *height map*.

L’échelle mésoscopique : la normale. L’échelle mésoscopique est une échelle intermédiaire entre les échelles macro- et microscopique, et à laquelle les détails sont à la limite d’être distinguables individuellement : c’est l’échelle du pixel. Dans notre cas, elle correspond aux variations des normales de la surface déplacée et sont stockées dans une *normal map*. De cette manière, alors que les variations géométriques de la surface sont sur-pixel, les variations de ses normales permettent d’enrichir l’image de détails variant d’un pixel à l’autre.

L’échelle microscopique : la BRDF. L’échelle microscopique est l’échelle à laquelle les détails ne sont plus distinguables individuellement : c’est l’échelle sous-pixel. Lorsqu’on s’éloigne suffisamment, les détails géométriques de la surface ne sont plus directement observables mais produisent collectivement des effets photométriques mesurables et modélisables par une *Bidirectional Reflectance Distribution Function (BRDF)*. La BRDF variant sur la surface est représentée par une *Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF)*.

Domaine de validité de ce modèle multi-échelles. Cette représentation multi-échelles surfacique peut servir à amplifier divers types de surfaces, comme celle de l’océan, celle d’un terrain ou encore celle d’un personnage. Ces utilisations typiques sont illustrées en figure 1.14. C’est donc une représentation générale qui ne fait pas de supposition sur le type de surface considéré, mais qui s’accompagne d’un certain nombre

d'hypothèses sur son application locale. Pour que l'application de ces textures soit valide, nous supposons que dans le voisinage qui correspond à l'espace projeté dans le même pixel :

- La géométrie basses-fréquences sur laquelle sont appliquées les textures est localement plane, ou, moins restrictif, à courbure faible et décorrélée du contenu des textures.
- La paramétrisation locale qui sert à appliquer les textures sur la surface ne varie pas localement. Elle peut être distordue, mais son jacobien est localement constant. C'est sous cette hypothèse que nous pouvons utiliser des textures pré-filtrées.
- Les directions d'observation ω_o et d'éclairage ω_i sont supposées localement constantes.

Ces trois hypothèses sont des contraintes fortes sur le domaine de validité de ce modèle. Elles apportent des connaissances a priori que nous pourrions exploiter (*Critère de choix 2*).

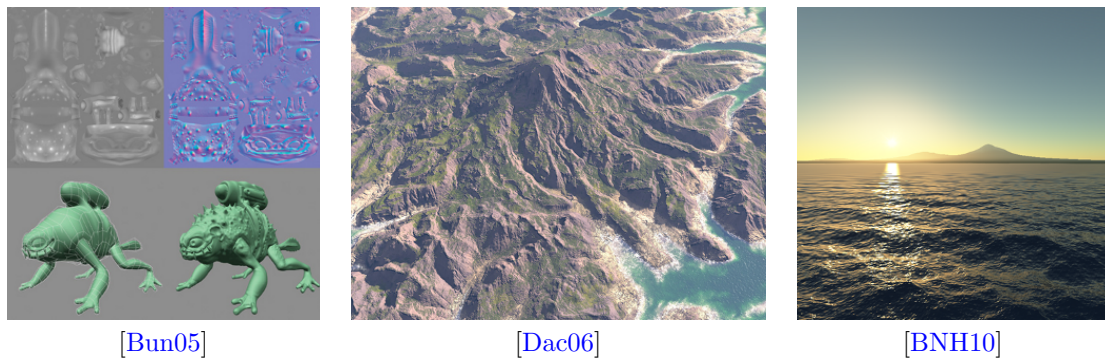


FIGURE 1.14 – Exemples de surfaces amplifiées avec des height maps.

Motivation industrielle. Les surfaces amplifiées sont parmi les primitives standard des pipelines de production. Elles conviennent parfaitement à la représentation des surfaces très détaillées à différentes étapes de la chaîne de production. Travailler avec des textures permet aux artistes d'éditer les détails en faisant abstraction de leur support géométrique. Cela constitue un gain de productivité important. En effet, les détails stockés dans des textures sont faciles à créer et à manipuler contrairement à la géométrie brute : il est bien plus simple pour les artistes de peindre directement dans la texture, comme on peindrait une image, que d'ajouter des triangles à la main. Le contenu d'une texture est aussi plus facile à modifier et à réutiliser. Certaines opérations simples comme le copier-coller sont faciles à faire avec des images mais seraient un casse-tête avec la géométrie en 3D. Il en va de même pour les opérations mathématiques dont même les plus simples, comme l'addition, sont mal définies en 3D.

Disney possède son propre format pour stocker les textures de matériaux et de hauteur : le format P-tex [BL08]. Sa souplesse et son efficacité en font un format utilisable comme format d'édition et de pré-visualisation pour les artistes, mais aussi comme format directement utilisable pour le rendu des images de production. En effet, le stockage et le traitement d'un maillage basse-résolution accompagné de textures haute-résolution est plus efficace que le traitement du maillage haute-résolution équivalent, en particulier en terme de mémoire. Un exemple de maillage P-tex amplifié avec des textures haute-résolution est donné en figure 1.15.

Analyse de complexité. Le format P-tex est abondamment utilisé en production. Par exemple, on peut voir le même dinosaure en arrière plan d'une scène du film *Wreck-it Ralph* dans la figure 1.16. Cette image est commode pour évaluer l'ordre de grandeur de la complexité de la scène. Le modèle du dinosaure, amplifié par sa height map à la résolution maximale, ferait environ 10^8 triangles et il recouvre environ 200×200 pixels dans cette image, ce qui nous fait un ratio de virtuellement 1000 triangles par pixel qu'il faudrait intégrer un par un avec un algorithme brute force.



FIGURE 1.15 – Le format *P-text* de Disney permet de paramétrer des maillages pour leur appliquer des textures de matériaux et de déplacement très haute-résolution [BL08].



FIGURE 1.16 – Une scène du film *Wreck-it Ralph* de Disney[©] réalisée avec des personnages au format *P-text*.

Pour gérer la complexité, les pipelines de production actuels utilisent déjà des height map filtrées (où c'est la fonction de hauteur qui est directement moyennée), combinées avec une subdivision des triangles qui adapte leur densité à leur distance à l'écran de manière à garder un ratio constant d'environ 1 triangle par pixel là où il y en avait virtuellement 1000 par pixel : la complexité est maîtrisée.

L'utilisation d'une height map offre une paramétrisation locale dans laquelle le signal géométrique est explicite : l'espace texture. Cela en fait une primitive particulièrement facile à filtrer, contrairement à de la géométrie brute. Mais pour garantir que la valeur du pixel soit correcte, il faut que l'apparence calculée avec un seul triangle soit identique à l'apparence qu'auraient eue les 1000 triangles à la résolution maximale. Modéliser et prédire l'apparence d'une surface amplifiée à différentes échelles est un problème peu traité dans la littérature.

Le problème des approches de filtrage de textures. Le cas de l’amplification des surfaces est particulièrement intéressant d’un point de vue historique. Les premiers algorithmes de rendu tournaient sur du hardware disposant de peu de puissance et peu de mémoire. En conséquence, les modèles 3D étaient très pauvres géométriquement, avec un faible nombre de triangles. Appliquer sur ces modèles des textures, qui contenaient divers attributs comme des matériaux ou des normales, était un moyen économique d’ajouter de la complexité. Ce paradigme de rendu “géométrie basse-résolution + textures” introduit plusieurs hypothèses implicites :

1. La résolution des textures est bien plus grande que celle des triangles sur lesquels elles sont appliquées. En conséquence, le contenu des textures n’est pas corrélé avec sa visibilité. En effet, puisqu’à l’intérieur du même triangle la courbure et la visibilité sont constantes, il n’y a pas de corrélations possibles entre les attributs de la texture et leur visibilité. C’est sous cette hypothèse qu’on peut filtrer séparément le contenu des textures et le masquage ou les ombres sur une partie de la surface.
2. Les contenus des différentes textures ne sont pas corrélés entre eux. C’est sous cette hypothèse qu’on peut, par exemple, filtrer séparément une texture de matériaux et une texture de normales, comme illustré dans la figure 1.17.

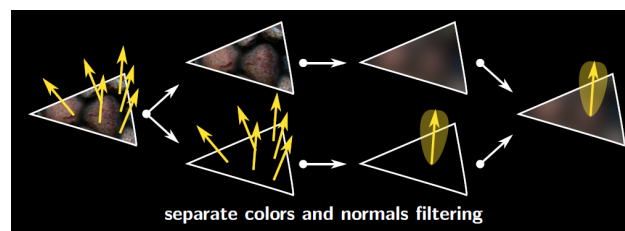


FIGURE 1.17 – Hypothèses implicites du filtrage de texture : 1. La taille des triangles est grande comparée à la résolution des textures. 2. Les contenus des textures sont supposés non corrélés.

Les modèles multi-échelles qui font ces hypothèses sont très étudiés dans la bibliographie. Bruneton et Neyret proposent une revue des différentes techniques de filtrage (linéaires ou non linéaires) dédiées aux différentes fonctions intervenant dans l’apparence des surfaces (les matériaux, les normales maps, la visibilité, etc.) [BN12a]. La revue pointe justement que ces approches sont souvent développées pour chacune des fonctions séparément, sous l’hypothèse de non corrélation entre les différentes fonctions. Il existe très peu de travaux qui tiennent compte de ces corrélations.

Pourtant, ces hypothèses ne sont plus vraies aujourd’hui, en particulier parce que la complexité des modèles a rattrapé celle des textures. Par exemple, dans le rendu de production, les pipelines du type REYES font un usage massif de la tessellation. Cela est vrai aussi pour le rendu temps-réel, puisqu’aujourd’hui la tessellation est implémentée en hardware dans les GPU modernes et fait directement partie des API standards comme OpenGL ou Direct3D. Un exemple de surface haute résolution est donné en figure 1.18.

Lorsque la géométrie a une complexité comparable à celle des textures, les hypothèses de séparabilité et de non corrélation ne tiennent plus dans le cas général. Par exemple, l’apparence d’une surface géométriquement complexe n’est pas l’apparence d’une surface plate avec des textures. En effet, les effets de visibilité qui sont négligés avec une normal map peuvent avoir une importance avec une height map. La figure 1.19 illustre la différence d’apparence entre ces deux techniques.

L’objet de cette thèse est d’attaquer le problème du filtrage de l’apparence des surfaces détaillées lorsque ces hypothèses ne sont plus valides. On souhaite mettre au point un modèle qui tienne compte des corrélations des attributs de la surface entre eux, ou avec la visibilité de la surface induite par l’existence de la géométrie déplacée.

Le problème des approches avec des BTFs. Dans la littérature, les travaux concernant le filtrage de l’apparence des surfaces en tenant compte des corrélations sont des méthodes qui utilisent des *Bidirectional texture function* (BTF) pour représenter l’échelle microscopique [BM93, MCT+05, WDR09]. Une BTF est

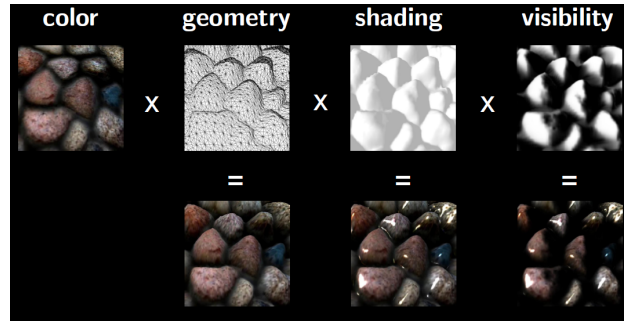


FIGURE 1.18 – La tessellation massive permet à la géométrie d’avoir une résolution comparable à celle des textures.

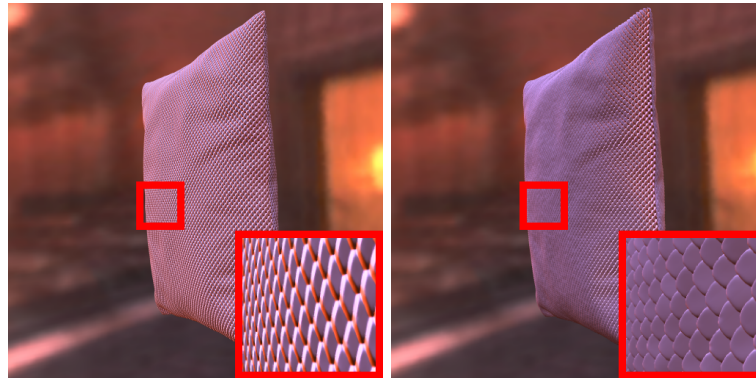


FIGURE 1.19 – En partant des mêmes données, l’apparence obtenue avec une height map (droite) n’est pas identique à celle obtenue avec une normal map (gauche) : avec une height map certains détails de la surface deviennent invisibles et ne contribuent plus à son apparence [DHI⁺ 13].

une texture 2D, définie sur l’ensemble de la surface, dont chaque texel contient une BRDF (4D) : c’est donc une fonction 6D. Elle représente l’apparence filtrée des échelles macroscopique et mésoscopique lorsque leur contenu est sous-pixel. En particulier, la technique de Becker et Max utilise un concept multi-échelles similaire à celui que nous décrivons [BM93]. Nous avons précédemment évoqué les approches “brute force” pour le calcul des intégrales du rendu. Dans le cas des BTFs, on peut parler d’approches “brute force pré-calcul et mémoire”. En effet, les méthodes de filtrage qui utilisent des BTFs les créent en pré-calculant la fonction 6D et en la stockant dans une texture : la partie “brute force” ne se situe plus dans le calcul du rendu, mais dans le pré-calcul et le stockage. Ces algorithmes sont donc scalables en terme de complexité du rendu et permettent de tenir compte des corrélations contrairement aux approches classiques de filtrage de textures. Mais ils ne sont pas scalables en terme de l’utilisation de la mémoire. En effet, le stockage des BTFs pose plusieurs problèmes pratiques car il faut trouver une représentation adaptée au stockage d’une fonction 6D et qui doit satisfaire plusieurs critères pour être utilisable en pratique :

- La représentation doit être capable de représenter précisément la fonction filtrée tout en ayant un coût de stockage raisonnable.
- La représentation doit pouvoir représenter des BRDFs basses-fréquences mais aussi hautes-fréquences. Des cas d’école simples sont des surfaces plates parfaitement diffuse (basses-fréquences) ou parfaitement spéculaire (fréquences infinies). Cette contrainte exclue d’emblée les techniques à décomposition dans une base (comme les harmoniques sphériques [HSRG07]) qui ne sont raisonnables que pour des fréquences limitées.
- La représentation doit être interpolable, à la fois spatialement et entre les niveaux de détails. En effet,

ne pas interpoler introduit de l'aliasing. A défaut d'avoir des représentations interpolables, certaines méthodes interpolent "à la main" la BTF après évaluation. Ce qui veut dire qu'il faut l'évaluer au moins 8 fois (4 fois spatialement et 2 fois entre les niveaux de détails). Dans le cas des silhouettes, le filtre appliqué sur la texture n'est pas isotrope et demande une reconstruction anisotrope souvent émulée avec plusieurs échantillons, ce qui peut faire passer le nombre d'évaluations de 8 fois à 16 ou 32 voire plus. Les représentations non interpolables reviennent donc très cher en calcul.

Trouver une représentation pour une fonction 6D qui satisfasse tous ces critères est un problème très difficile. Nous pensons que cette approche n'est pas efficace, car elle cherche à traiter l'ensemble des données sans utiliser les connaissances a priori sur le problème qu'on cherche à résoudre : c'est pour ça que nous la qualifions de "brute force". Intuitivement, la fonction 6D à reconstruire est la composition de l'information présente dans la représentation par celle présente dans le modèle :

$$\text{modèle}(\text{représentation}) = \text{fonction 6D}.$$

Pour le dire simplement, plus le modèle contient d'information, moins la représentation a besoin d'en apporter. Vu sous cet angle, le problème des BTFs est que le modèle est tout simplement inexistant : toute l'information est contenue uniquement dans la représentation et c'est pour cette raison qu'elle est coûteuse et difficile à mettre en oeuvre.

Notre approche consiste au contraire à aller le plus loin possible sur la partie modélisation en exploitant au maximum les connaissances a priori sur le modèle, de manière à pouvoir utiliser une représentation minimaliste tout en garantissant une reconstruction précise de la fonction 6D. Nous avons vu que le modèle des surfaces détaillées pose un cadre assez spécifique avec des propriétés de localité à exploiter. Il s'agit donc de trouver un cadre théorique efficace pour les formuler.

La théorie des micro-facettes comme cadre théorique. Certains modèles de BRDF sont proches de notre problème. En particulier, les BRDFs dites à *micro-facettes*, sont celles qui modélisent la réflectance de height maps microscopiques. Ceci est très proche de notre modèle multi-échelles basé sur les height maps. Le modèle à micro-facettes suppose que la macro-surface visible est composée d'une infinité de micro-facettes, trop petites pour être distinguables, et qui forment une height map d'amplitude microscopique, appelée la micro-surface. La BRDF est modélisée en calculant la réflectance de la height map microscopique, en tenant compte de la réflectance des micro-facettes individuelles, de leur distribution statistique, de leur visibilité etc. La théorie des micro-facettes constitue un modèle de physique statistique de la réflectance des surfaces et dont l'étude contient des outils théoriques et des dérivations que nous pourrions réutiliser pour notre étude.

Bien qu'elle constitue un excellent point de départ, la théorie des micro-facettes n'est pas pour autant directement utilisable pour nos besoins. En effet, elle ne modélise actuellement que les échelles macro- et microscopique et non l'échelle mésoscopique à l'interface entre les deux : ce n'est pas une théorie multi-échelles mais seulement *bi-échelles*. En conséquence, elle ne permet pas de formuler un modèle multi-échelles avec une transition continue entre les différentes échelles. Le chapitre 5 est dédié à étendre la théorie des micro-facettes pour en faire une théorie multi-échelles utilisable pour notre problème.

D'autre part, nous verrons aussi que la littérature sur la théorie des micro-facettes laisse certaines questions importantes en suspens, notamment concernant ses modèles de micro-surfaces. Certains travaux l'utilisent parfois comme une boîte noire pour créer de nouveaux modèles de BRDFs, sans pour autant comprendre, ou au moins discuter, toutes leurs propriétés. Mais il est indispensable de les comprendre si on souhaite les utiliser comme point de départ pour concevoir un modèle multi-échelles de qualité. Avant de concevoir une extension multi-échelles, nous proposons donc de commencer par revisiter la théorie dans son état actuel pour comprendre ses propriétés et en découvrir de nouvelles, car, comme nous l'avons vu, plus le modèle contient de propriétés compréhensibles et exploitables, plus il est efficace. C'est l'objet du chapitre 4.

1.3.3 Objectif formel de la thèse

Dans cette section, nous définissons de manière formelle l’objectif que nous poursuivons dans cette thèse.

L’équation d’illumination locale. L’équation du filtrage de l’apparence des surfaces multi-échelles, sous les hypothèses de localité listées plus haut est donnée formellement par l’équation d’illumination locale :

$$L(\omega_o) = \frac{\int_{\Omega_i} \int_{\mathcal{M}} L(\omega_i, p_m) \rho_{\mathcal{M}}(p_m, \omega_o, \omega_i, \omega_m(p)) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1(\omega_o, p_m) G_1(\omega_i, p_m) k(p_m) dp_m d\omega_i}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle G_1(\omega_o, p_m) k(p_m) dp_m},$$

où \mathcal{M} est la zone de la surface recouverte par l’empreinte du pixel, dont le filtre est $k(p_m)$ et p_m sa paramétrisation spatiale. Chaque point p_m de la surface a une normale $\omega_m(p_m)$. La contribution d’une micro-facette dépend aussi de sa surface projetée dans la direction d’observation $\langle \omega_o, \omega_m(p_m) \rangle$. L’irradiance incidente au point p_m , donnée par $L(\omega_i, p_m) \langle \omega_i, \omega_m(p_m) \rangle$, est multipliée par la BRDF $\rho_{\mathcal{M}}(p_m, \omega_o, \omega_i, \omega_m(p_m))$ en ce point et contribue au pixel seulement si le point n’est ni masqué $G_1(\omega_o, p_m)$, ni ombré $G_1(\omega_i, p_m)$. Comme la radiance est une mesure de densité d’énergie, il faut renormaliser l’intégrale par la surface projetée dans le pixel. C’est ce que calcule l’intégrale au dénominateur. Pour alléger la notation, nous réécrivons cette intégrale où la variable p_m est présente de manière implicite dans chacune des fonctions :

$$L(\omega_o) = \frac{\int_{\Omega_i} \int_{\mathcal{M}} L(\omega_i) \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m d\omega_i}{\int_{\mathcal{M}} \langle \omega_o, \omega_m \rangle G_1(\omega_o) dp_m}. \quad (1.1)$$

A noter que nous ferons toujours abstraction du filtre $k(p_m)$ associé à l’empreinte du pixel, sans perte de généralité. En effet, il est toujours possible de se ramener à une formulation sans cette fonction en opérant un changement de variable $dp'_m = k(p_m) dp_m$.

D’un problème d’intégration 4D à un problème d’intégration 2D. L’équation (1.1) décrit une intégration dans un espace 4D : il y a deux dimensions spatiales (la variable p_m qui décrit le patch de surface \mathcal{M}) et deux dimensions directionnelles (la variable ω_i de la direction incidente). Notre objectif est d’utiliser un modèle multi-échelles de surface pour transformer ce problème d’intégration 4D en un problème d’intégration 2D. En effet, l’équation peut s’exprimer comme une intégration bidimensionnelle en écrivant :

$$L(\omega_o) = \int_{\Omega_i} L(\omega_i) \bar{\rho}(\omega_o, \omega_i) (\omega_g \cdot \omega_i) d\omega_i, \quad (1.2)$$

où $\bar{\rho}$ est la BRDF représentant le matériau multi-échelles associé à la surface filtrée :

$$\bar{\rho}(\omega_o, \omega_i) (\omega_g \cdot \omega_i) = \frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m \rangle G_1(\omega_o) dp_m}. \quad (1.3)$$

Nous voyons qu’utiliser une description multi-échelles de la surface permet de faire tomber les deux dimensions spatiales du problème. De cette manière, la complexité du rendu ne dépend plus de la complexité spatiale de la scène : c’est l’objectif principal que nous cherchons à atteindre dans cette thèse. En ce qui concerne la complexité de l’éclairage, nous verrons différentes stratégies pour l’intégrer efficacement avec un cahier des charges temps-réel/GPU ou dans le contexte du rendu offline dans un moteur de type Monte Carlo avec de l’importance sampling.

Objectifs. En résumé, nos objectifs sont les suivants :

1. Mettre au point un modèle théorique de la BRDF ρ émergente de l’équation (1.3) qui donne l’apparence de la surface multi-échelles, et analyser les degrés de liberté et les corrélations entre chacun des termes de l’intégrale de manière à trouver des approximations efficaces et précises pour la représenter efficacement en pratique.
2. Mettre au point des modèles de calcul pour intégrer l’éclairage incident donné par l’équation (1.2).

1.3.4 Description formelle de la thèse

Dans cette section, nous expliquons formellement l'organisation et le contenu des différents chapitres et expliquons comment ils s'enchaînent et contribuent à atteindre les objectifs définis précédemment.

La théorie standard des micro-facettes. Le chapitre 4 commence par introduire la propriété fondamentale de la conservation de l'aire projetée visible :

$$\int_{\mathcal{M}} \langle \omega_o, \omega_m \rangle G_1(\omega_o) dp_m = |\omega_g \cdot \omega_o|. \quad (1.4)$$

Cette propriété nous sert de point de départ pour dériver de manière unifiée l'ensemble des résultats de la théorie des micro-facettes. En l'introduisant dans l'équation (1.3), nous obtenons :

$$\begin{aligned} \rho &= \frac{1}{|\omega_g \cdot \omega_i|} \frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m \rangle G_1(\omega_o) dp_m} \\ &= \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m. \end{aligned}$$

Le chapitre 4 dérive les autres résultats classiques de la théorie des micro-facettes en introduisant deux hypothèses dont nous essayerons de nous passer par la suite. La première est que, dans les BRDFs à micro-facettes, le matériau est supposé constant, ce qui revient à dire qu'il est moyenné sur la surface sans tenir compte des corrélations avec les autres termes de l'intégrale :

$$\begin{aligned} \rho &= \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m \\ &\approx \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\mathcal{M}} \left(\frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) dp_m}{\int_{\mathcal{M}} dp_m} \right) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_o) dp_m. \end{aligned}$$

Dans un second temps, nous verrons que pour obtenir des formes analytiques, les modèles à micro-facettes font certaines hypothèses sur la micro-surface. L'hypothèse fondamentale du modèle de micro-surface de Smith [Smi67], que nous choisirons d'utiliser, est que le masquage $G_1(\omega_o)$ et l'ombrage $G_1(\omega_i)$ ne sont pas corrélés à l'orientation des normales qui interviennent dans les produits scalaires et dans la BRDF. Ils peuvent donc être moyennés à part. Par ailleurs, $G_1(\omega_o)$ et $G_1(\omega_i)$ sont aussi supposées indépendantes dans les modèles existants. Au final, l'équation standard des modèles de BRDFs à micro-facettes est donnée par :

$$\begin{aligned} \rho &\approx \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\mathcal{M}} \left(\frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) dp_m}{\int_{\mathcal{M}} dp_m} \right) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_o) dp_m \\ &\approx \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\mathcal{M}} \left(\frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) dp_m}{\int_{\mathcal{M}} dp_m} \right) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle dp_m \frac{\int_{\mathcal{M}} G_1(\omega_o) dp_m}{\int_{\mathcal{M}} dp_m} \frac{\int_{\mathcal{M}} G_1(\omega_o) dp_m}{\int_{\mathcal{M}} dp_m}. \end{aligned}$$

Théorie multi-échelles des micro-facettes. Dans le chapitre 5, nous expliquons que la théorie standard des micro-facettes est une théorie *bi-échelles* qui ne modélise que les échelles macro- et microscopiques et ne permet pas de transitions continues entre les deux. Pour obtenir une théorie multi-échelles, et donc permettre des transitions entre les niveaux de détails, nous introduisons une nouvelle échelle : l'échelle *mésoscopique*. En revanche, c'est sous les hypothèses du modèle bi-échelles qu'est dérivée l'équation (1.4) de la conservation de l'aire projetée visible. Celle-ci est invalidée par l'introduction de l'échelle mésoscopique et il en va de même pour tous les résultats précédents. Nous montrons que dans le modèle multi-échelles, l'équation (1.4) s'exprime de manière correcte sous une forme généralisée :

$$\int_{\mathcal{M}} \langle \omega_o, \omega_m \rangle G_1(\omega_o) dp_m = \frac{|\omega_o \cdot \omega_{\bar{m}}|}{|\omega_g \cdot \omega_{\bar{m}}|},$$

où $\omega_{\bar{m}}$ est la *méso-normale* et ω_g la *macro-normale*. Cette équation montre que la conservation de l'aire projetée dépend de la projection de la méso-surface dans la direction d'observation, mais aussi de son décalage par rapport à la macro-surface. De la même manière, l'irradiance arrivant sur la méso-surface n'est pas $\langle \omega_g, \omega_i \rangle$ mais $\langle \omega_{\bar{m}}, \omega_i \rangle$. Cela nous permet de réécrire l'équation exacte :

$$\begin{aligned} \rho &= \frac{1}{|\omega_i \cdot \omega_{\bar{m}}|} \frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m \rangle G_1(\omega_o) dp_m} \\ &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m, \end{aligned}$$

ainsi que l'approximation précédente du modèle standard, qui, dans le modèle multi-échelles, devient :

$$\rho \approx \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \left(\frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) dp_m}{\int_{\mathcal{M}} dp_m} \right) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle dp_m \frac{\int_{\mathcal{M}} G_1(\omega_o) dp_m}{\int_{\mathcal{M}} dp_m} \frac{\int_{\mathcal{M}} G_1(\omega_i) dp_m}{\int_{\mathcal{M}} dp_m}.$$

Avec cette forme introduite dans le chapitre 5, nous savons désormais comment définir des transitions continues entre les échelles et sommes bien partis pour commencer la mise au point d'un modèle multi-échelles. En revanche, nous avons hérité des approximations du modèle standard : le matériau est supposé constant et nous supposons qu'il n'y a pas de corrélation entre $G_1(\omega_o)$ et $G_1(\omega_i)$. Les chapitres suivants sont dédiés à se soustraire au maximum à ces hypothèses.

Représentation de la surface. Dans le chapitre 7, nous définissons un cahier des charges pour les applications que nous proposons de mettre au point. Dans le chapitre 8, nous choisissons une représentation pour la surface multi-échelles qui satisfait ce cahier des charges : nous expliquons pourquoi notre choix se porte sur des surfaces à statistiques gaussiennes et nous montrons comment la représentation de ces statistiques est compatible avec l'animation de la surface.

Les chapitres évoqués par la suite font tous l'hypothèse d'une surface à statistiques gaussiennes.

Corrélation entre l'ombrage et le masquage. Dans le chapitre 6, nous proposons un modèle qui tient compte des corrélations possibles entre le masquage $G_1(\omega_o)$ et l'ombrage $G_1(\omega_i)$ qui sont moyennés séparément dans le modèle standard. En effet, lorsque les directions d'observation ω_o et ω_i sont proches, les visibilitées pour les deux directions deviennent corrélées et ne peuvent pas se moyenner séparément. Par exemple, lorsqu'une scène de nuit est éclairée par un observateur avec une lampe frontale, les ombres ne sont pas visibles, car ce qui est visible pour la lumière l'est aussi pour l'observateur : c'est ce qu'on appelle l'effet *hotspot*. Concrètement, nous souhaitons modéliser :

$$\begin{aligned} \rho &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m \\ &\approx \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \left(\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) dp_m \right) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle dp_m \underbrace{\frac{\int_{\mathcal{M}} G_1(\omega_o) dp_m}{\int_{\mathcal{M}} dp_m} \frac{\int_{\mathcal{M}} G_1(\omega_i) dp_m}{\int_{\mathcal{M}} dp_m}}_{\frac{\int_{\mathcal{M}} G_1(\omega_o) G_1(\omega_i) dp_m}{\int_{\mathcal{M}} dp_m}}, \end{aligned}$$

où nous avons remplacé le produit des intégrales $\int_{\mathcal{M}} G_1(\omega_o) dp_m \int_{\mathcal{M}} G_1(\omega_i) dp_m$ par l'intégrale du produit $\int_{\mathcal{M}} G_1(\omega_o) G_1(\omega_i) dp_m$. Dans le chapitre 6, nous proposons une forme analytique pour l'évaluation de la fonction de masquage et d'ombrage sous cette forme jointe.

Corrélations avec les matériaux. Dans le chapitre 9, nous étudions les cas où le matériau de la surface multi-échelles n'est pas constant et en particulier ceux où il est corrélé avec la visibilité ou avec les normales de la surface et ne peut pas être intégré de manière séparable. Nous verrons qu'il y a deux types de corrélations fondamentales, données par les **normales** et les **fonctions de masquage et d'ombrage** :

$$\rho = \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m$$

Dans le cas où le matériau est corrélé avec la visibilité, nous avons :

$$\begin{aligned}\rho &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m \\ &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \left(\frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) G_1(\omega_o) G_1(\omega_i) dp_m}{\int_{\mathcal{M}} dp_m} \right) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle dp_m,\end{aligned}$$

où $\int_{\mathcal{M}} \rho(\omega_o, \omega_i, \omega_m) G_1(\omega_o) G_1(\omega_i) dp_m$ est le matériau moyen visible simultanément depuis ω_o et ω_i . Nous montrons que ces matériaux corrélés avec la visibilité peuvent se représenter comme une *color map* appliquée sur les hauteurs de la micro-surface et nous proposons une approximation analytique pour l'évaluation de cette moyenne visible. Lorsque le matériau est corrélé avec les normales de la micro-surface, nous proposons d'approximer :

$$\begin{aligned}\rho &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o) G_1(\omega_i) dp_m \\ &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \left(\frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle dp_m}{\int_{\mathcal{M}} dp_m} \right) G_1(\omega_o) G_1(\omega_i) dp_m.\end{aligned}$$

Nous montrons que ces matériaux corrélés avec les normales peuvent se représenter comme une *color map* appliquée sur les normales de la micro-surface et nous proposons une manière efficace d'approximer cette intégrale.

Intégration de l'éclairage incident. Les chapitres présentés jusqu'ici ont pour but la représentation et l'approximation de la BRDF ρ . Les derniers chapitres sont destinés à calculer son intégrale avec l'éclairage incident, c'est-à-dire à calculer l'équation (1.2).

Le chapitre 10 propose des techniques d'intégration numériques compatibles avec un cahier des charges temps-réel, tandis que le chapitre 11 propose une nouvelle technique d'*importance sampling* (échantillonnage par importance) destinée à évaluer cette intégrale dans un moteur de rendu offline de type Monte Carlo.

1.3.5 Organisation des chapitres

Voici les principales contributions scientifiques et techniques présentées dans chacun des chapitres.

Chapitre 4

- Nous proposons une dérivation exacte du terme de masquage pour les BRDFs à micro-facettes. Cette dérivation montre formellement quel est le lien entre ce terme et la normalisation correcte de la BRDF, lien qui était inconnu de la communauté auparavant. Cette dérivation permet de mieux comprendre le rôle que jouent les différents termes du modèle et d’unifier des résultats existants.
- Nous en déduisons un simple test de normalisation, que nous appelons le *Weak White Furnace Test* que doivent vérifier les modèles à micro-facettes mathématiquement bien posés.
- Nous montrons les propriétés d’invariance du modèle, qui peuvent être utilisées pour dériver de manière simple des résultats pour des micro-surfaces anisotropes lorsque l’anisotropie est due à des étirements de la micro-géométrie.

Chapitre 5

- Nous introduisons l’échelle mésoscopique dans la théorie des micro-facettes pour en faire une théorie multi-échelles et non uniquement bi-échelles. En pratique, nous montrons que cela revient à la rendre compatible avec des distributions non centrées en 0, c’est-à-dire avec un décalage entre les échelles méso- et macroscopique.

Chapitre 6

- Nous proposons une étude du phénomène de corrélation entre les fonctions de masquage et d’ombrage, aussi appelé le “hotspot”, toujours négligé dans les modèles de BRDF précédents.

Chapitre 8

- Nous montrons comment la représentation de LEAN Mapping [OB10] peut être rendue compatible avec l’animation des surfaces et donc de la micro-géométrie, ce qui évite d’avoir à réinitialiser la représentation pour chaque image d’une séquence animée.

Chapitre 9

- Nous proposons une solution au problème de filtrage des textures procédurales color mappées.
- Nous montrons comment cette solution peut être utilisée lorsque les color maps sont appliquées sur des descripteurs géométriques de la micro-surface et deviennent corrélées avec la visibilité.

Chapitre 10

- Nous dérivons un schéma d’intégration numérique inspiré de *filtered importance sampling* pour intégrer la BRDF du modèle multi-échelles avec l’éclairage avec des performances temps-réel.

Chapitre 11

- Nous proposons un nouveau schéma d’importance sampling pour les modèles à micro-facettes, compatible avec les BRDFs multi-échelles présentées dans les autres chapitres, mais aussi avec des modèles qui modélisent la réfraction (BTDFs), et utilisable dans un moteur de rendu offline de type Monte Carlo.

1.3.6 Organisation des annexes

A la fin du manuscrit se trouvent des chapitres annexes. Ceux-ci sont en anglais et sont constitués de *supplemental materials* (documents supplémentaires) attachés à mes publications. En effet, j’ai accompagné la plupart de mes articles de documents supplémentaires, dont la longueur dépasse parfois celle des articles eux-mêmes.

Pourquoi des supplemental materials ? Si la soumission aux conférences permet un retour assez rapide et une visibilité importante en cas d’acceptation, je trouve en revanche que le format standard de publication dans les conférences de l’informatique graphique (articles de 8-10 pages) n’est pas toujours adapté à produire des articles de qualité, en terme de :

1. Validation. Chaque résultat montré dans un article a un rôle spécifique à jouer concernant la validation de la technique proposée. Travailler sur des problématiques de synthèse d’images rend parfois tentante la validation par l’image. Mais pour valider un algorithme rigoureusement, il faut explorer de manière exhaustive l’espace de ses paramètres d’entrée. Pour le faire proprement, il est important de trouver comment décomposer cet espace pour obtenir des séries de *tests unitaires* qu’il faut ensuite réaliser et valider. Malheureusement, le format de 8-10 pages n’offre pas la possibilité de montrer de tels résultats de manière exhaustive. Un document supplémentaire peut servir de support pour montrer ce genre de validation.
2. Reproductibilité. Il existe souvent une différence entre la théorie et la pratique. Expliquer en théorie comment fonctionne un algorithme n’est parfois pas suffisant pour que le lecteur puisse le reproduire facilement (par exemple lorsque l’algorithme utilise certaines variables “magiques” qu’il faut régler à la main). Un document supplémentaire peut expliquer plus en détail comment certaines difficultés techniques ou certains choix arbitraires ont été réalisés et ainsi permettre de reproduire plus facilement les résultats montrés dans l’article.
3. Maximisation de l’impact. Si l’implémentation de l’algorithme dans un langage de programmation spécifique n’a pas sa place dans un article, partager le code permet de rendre son utilisation immédiate et maximise son impact. Partager le code n’est pas toujours possible et certains types d’articles s’y prêtent moins que d’autres. Mais les implémentations des articles dédiés aux modèles de BRDF et de shading que nous étudions dans cette thèse tiennent parfois en une seule fonction analytique. Dans ce cas, il n’y a aucune excuse à ne pas partager le code dans un document supplémentaire.

Si les documents supplémentaires sont une manière d’enrichir un article, ils ne doivent pas être utilisés comme une manière détournée d’y ajouter des pages. L’ensemble des idées et de la méthodologie doivent se trouver dans l’article lui-même qui doit toujours pouvoir être lu, compris et reproduit de manière autonome. Le document supplémentaire a pour seule vocation de présenter des instances possibles des idées et de la méthodologie présentées dans l’article. Par exemple, pour la validation, les critères à valider et la méthodologie à adopter pour le faire doivent être expliqués dans l’article et celui-ci doit présenter au moins un exemple d’un tel test de validation. Par contre, l’énumération exhaustive de ces tests peut être mise dans le document supplémentaire.

Ci-dessous se trouve la liste des annexes, leurs liens avec les chapitres et les publications.

Annexe A associée au chapitre 4 et à l'article [Hei14]

- Nous y détaillons notre dérivation de la fonction de masquage de Smith en partant de la conservation de l'aire projetée.
- Nous y partageons notre implémentation MATLAB pour reproduire le *Weak White Furnace Test*.

Annexe B associée au chapitre 6 et à l'article [HBP13]

- Nous y partageons notre implémentation MATLAB pour le calcul du facteur de corrélation r_0 .

Annexe C associée au chapitre 10 et à l'article [DHI+13]

- Nous y détaillons les dérivations des jacobiens liés aux transformations entre les différents espaces utilisés en théorie des micro-facettes.

Annexe D associée au chapitre 10 et à l'article [DHI+13]

- Nous y proposons une série de tests unitaires pour valider la convergence de la technique d'intégration numérique de la BRDF multi-échelles que nous proposons.

Annexe E associée au chapitre 9 et aux articles [HNPN13a] et [HNPN13b]

- Nous y détaillons le calcul des moments de la distribution de pentes visibles et nous y partageons notre implémentation C++ pour les calculer.

Annexe F associée au chapitre 11 et à l'article [HD14]

- Nous y détaillons l'implémentation de la technique d'importance sampling pour les cas particuliers des distributions de Beckmann et GGX.
- Nous y proposons des séries de tests de convergence en graphes et en images.

1.3.7 Communications associées à la thèse

Journaux et conférences avec comité de lecture

- [HD14] *Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals*. Eurographics Symposium on Rendering (**EGSR**), 2014.
Eric Heitz and Eugene d'Eon.
1st Best Paper Award.
- [Hei14] *Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs*. The Journal of Computer Graphics Techniques (**JCGT**), 2014.
Eric Heitz.
- [DHI+13] *Linear Efficient Antialiased Displacement and Reflectance Mapping*. ACM Transaction on Graphics (**SIGGRAPH Asia**), 2013.
Jonathan Dupuy*, **Eric Heitz***, Jean-Claude Iehl, Pierre Poulin, Fabrice Neyret, Victor Ostromoukhov.
***Joint first authors.**
- [HBP13] *Correlation Effect Between Transmitter and Receiver Azimuthal Directions on the Illumination Function from a Random Rough Surface*. Waves in Random and Complex Media (**WRCM**), 2013.
Eric Heitz, Christophe Bourlier, Nicolas Pinel.
- [HNPN13b] *Filtering Non-Linear Transfer Functions on Surfaces*. IEEE Transactions on Visualization and Computer Graphics (**TVCG**), 2013.
Eric Heitz, Derek Nowrouzezahrai, Pierre Poulin, Fabrice Neyret.
- [HNPN13a] *Filtering Color Mapped Textures and Surfaces*. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (**I3D**), 2013.
Eric Heitz, Derek Nowrouzezahrai, Pierre Poulin, Fabrice Neyret.
1st Best Paper Award.
- [HN12] *Representing Appearance and Pre-filtering Subpixel Data in Sparse Voxel Octrees*. High-Performance Graphics (**HPG**) 2012.
Eric Heitz and Fabrice Neyret.
1st Best Paper Award.

Cours

- *Physically Based Shading in Theory and Practice*. SIGGRAPH 2014 Course.
Stephen Hill, Stephen McAuley, Jonathan Dupuy, Yoshiharu Gotanda, **Eric Heitz**, Naty Hoffman, Sébastien Lagarde, Anders Langlands, Ian Megibben, Farhez Rayani, Charles de Rousiers.

1.3.8 Collaborations académiques et industrielles associées à la thèse

Dans la partie I de ce manuscrit, nous proposons essentiellement des études et des résultats théoriques liés à la théorie des micro-facettes. C'est un domaine dans lequel la littérature est pluridisciplinaire : il existe des travaux pertinents aussi bien dans la communauté de l'informatique graphique que dans celle de la physique. Pour maîtriser au mieux ce domaine, j'ai cherché à rencontrer et à discuter avec des experts des deux domaines, ce qui m'a amené à effectuer plusieurs collaborations académiques.

Par ailleurs, pour mettre au point des technologies utilisables en pratique et dans l'air du temps (le contenu de la partie II du manuscrit), j'ai également cherché à confronter mes idées et mes travaux avec les attentes du monde industriel d'aujourd'hui, ce qui m'a amené à effectuer deux séjours en entreprise (NVIDIA et Weta Digital) et à collaborer sur plusieurs projets avec des professionnels du jeu vidéo ou du rendu de production pour les films.

Voici une liste de ces projets et des différentes personnes avec qui je les ai réalisés.

Collaborations académiques

- J'ai candidaté et obtenu une bourse Explora'doc pour financer un séjour de 6 mois à l'Université de Montréal dans le but de collaborer avec **Pierre Poulin** et **Derek Nowrouzezahrai**. Nous avons travaillé sur une représentation multi-échelles des matériaux surfaciques lorsque ceux-ci sont corrélés à la visibilité. Ce projet a abouti à un article I3D [HNP13a] et sa version journal étendue [HNP13b]. Le contenu de ces articles fait l'objet du chapitre 9.
- Collaboration avec **Jonathan Dupuy**, doctorant en cotutelle avec le Laboratoire d'InfoRmatique en Image et Systèmes d'information et l'Université de Montréal, pour mettre au point le projet LEADR Mapping que nous avons publié à SIGGRAPH Asia [DHI+13]. Il recouvre l'extension multi-échelles de la théorie des micro-facettes présentée dans le chapitre 5 ainsi que les techniques d'intégration temps-réel présentées dans le chapitre 10.
- Collaboration avec **Christophe Bourlier** et **Nicolas Pinel**, physiciens dans le laboratoire IETR (Institut d'Electronique et des Télécommunications de Rennes). Je les avais initialement contactés pour leur demander des retours sur mon étude de la corrélation azimuthale entre le masquage et l'ombrage dans le modèle de Smith. Nous avons ensuite finalisé cette étude ensemble et l'avons publiée dans le journal de physique Waves in Random and Complex Media [HBP13]. Elle est présentée dans le chapitre 6.
- Collaboration pour la préparation du cours SIGGRAPH 2014 sur le *physically based shading* avec, en particulier, **Stephen Hill**, **Naty Hoffman**, **Sébastien Lagarde** qui travaillent dans le domaine du rendu temps-réel pour le jeu vidéo. Ce sont eux aussi qui m'ont encouragé à rédiger et à mettre au clair mes intuitions sur la fonction de masquage dans les modèles de BRDFs à micro-facettes. Grâce à leurs encouragements et à leur aide, ce document, qui avait pour but initial de n'être qu'un rapport de recherche, a finalement été accepté pour publication à the Journal of Computer Graphics Techniques [Hei14]. Il fait l'objet du chapitre 4.
- Collaboration avec **Eugene d'Eon**. Nous sommes entrés en contact au département Research de Weta Digital et nous avons continué à travailler ensemble à mon retour sur la technique d'importance sampling pour les BSDFs à micro-facettes. Ce travail a abouti à un article EGSR [HD14] et il est présenté dans le chapitre 11.

Collaborations industrielles

- Séjour de 3 mois à **NVIDIA** (Paris) avec **Cyril Crassin** et **David Luebke** pour représenter des niveaux de détails dans des volumes (voxels) et l’animation de la représentation. Ce travail a été développé avec la même méthodologie que mes travaux de thèse (modélisation physique de la fonction de phase, choix d’une représentation avec un cahier des charges, algorithmes d’évaluation, etc.), mais adaptée au cas des volumes plutôt qu’à celui des surfaces.
- Séjour de 1 mois à **Weta Digital** (Wellington, Nouvelle-Zélande) avec **Antoine Bouthors** dans leur département Shading. Le but de cette visite était, d’une part, d’aider Weta à intégrer la représentation LEADR Mapping [DHI⁺13] à leur pipeline de production. Cette représentation est aujourd’hui totalement intégrée à leur pipeline et fait partie de leurs primitives standard pour le contenu des textures. D’autre part, nous avons commencé un nouveau projet lié à la représentation et l’évaluation des niveaux de détails avec une représentation volumique. Ce projet, très prometteur, rejoint celui commencé à NVIDIA avec Cyril Crassin et il est toujours en cours de développement.

1.3.9 Impact mesurable à court terme des travaux associés à la thèse

Si les travaux présentés dans cette thèse ont déjà reçu plusieurs retours positifs de manière informelle, il est difficile d’anticiper aujourd’hui l’impact qu’ils auront à long terme. Voici néanmoins quelques éléments objectifs, que nous listons ci-dessous, de leur réception et de leur diffusion dans la communauté du rendu.

- Notre étude de la fonction de masquage dans les BRDFs à micro-facettes présentée dans le chapitre 4 et dans l’article [Hei14] sera citée dans **Physically Based Rendering, Third Edition** dont la sortie est prévue en 2015 (discussion privée avec **Matt Pharr**).
- La technique d’importance sampling des BSDFs à micro-facettes, que nous proposons dans le chapitre 11 et dans l’article [HD14], est implémentée aujourd’hui chez **Weta Digital**, chez **Sony** (d’après **Larry Gritz**), chez **Solid Angle** (discussion privée avec **Iliyan Georgiev**) et dans **Cycles**, le moteur de rendu de Blender (discussion privée avec **Brecht Van Lommel**). Elle fera partie intégrante de la prochaine distribution de **Mitsuba** (discussion privée avec **Wenzel Jakob**) et de **PBRT** (discussion privée avec **Matt Pharr**), qui sont des logiciels de référence pour le rendu *physically based*.
- La représentation et la BRDF multi-échelles de LEADR Mapping que nous proposons [DHI⁺13] sont utilisées chez **Weta Digital**.

Première partie

Revisiter la théorie des micro-facettes

Chapitre 2

Rappels généraux sur les BRDFs

Dans ce chapitre, nous rappelons les quantités physiques associées au transport de la lumière ainsi que la définition de la BRDF.

2.1 Quantités physiques

Les interactions entre la lumière et les surfaces sont décrites par des équations qui établissent des relations entre des quantités physiques. On les appelle souvent les équations du transport radiatif. Nous rappelons celles qui sont importantes pour la définition de la BRDF et elles sont résumées dans la table 2.1.

Flux radiant. Le flux radiant Φ est la quantité d'énergie qui passe à travers une surface par unité de temps. Il est homogène à une puissance exprimée en Watt (W).

Radiance. La radiance L est la densité de flux radiant Φ par unité de surface dA et par unité d'angle solide $d\omega_i$:

$$L(\omega_i, p) = \frac{\partial^2 \Phi}{\cos \theta_i \partial \omega_i \partial A}.$$

Réciproquement, le flux est l'intégrale de la radiance sur la surface et sur l'ensemble des directions :

$$\Phi = \int_A \int_{\Omega} L(\omega_i, p) \langle \omega_i, \omega_g \rangle d\omega_i dp.$$

La radiance est mesurée en Watt par stéradian par mètre carré ($\frac{W}{m^2 sr}$). Concrètement, dans une image, la radiance est l'unité dans laquelle sont exprimées les valeurs des pixels.

Irradiance. L'irradiance E est la densité de puissance arrivant sur la surface. C'est la somme du flux radiant Φ arrivant de toutes les directions, par unité de surface A :

$$E(A) = \frac{\partial \Phi}{\partial A}.$$

Elle peut aussi s'exprimer comme l'intégrale de la radiance dans toutes les directions :

$$E(p) = \int_{\Omega} L(\omega_i, p) \langle \omega_i, \omega_g \rangle d\omega_i.$$

L'irradiance est mesurée en Watt par mètre carré ($\frac{W}{m^2}$).

Définition	Nom	Unité
Φ	radiant power	Watt W
$L(\omega_i, A) = \frac{\partial^2 \Phi}{\cos \theta_i \partial \omega_i \partial A}$	radiance	$\frac{W}{m^2 sr}$
$E(A) = \frac{\partial \Phi}{\partial A}$	Irradiance	$\frac{W}{m^2}$

TABLE 2.1 – Notations : quantités physiques.

2.2 Configuration géométrique

L'ensemble des équations que nous utiliserons par la suite décrivent les échanges radiatifs entre une surface et deux directions. Nous les exprimerons toujours dans le même repère. Les notations sont résumées dans la table 2.2.

Paramétrisation de l'espace. Nous nous plaçons dans un repère orthonormé (x, y, z) qui donne une paramétrisation cartésienne de l'espace 3D. Pour les directions, nous utilisons des coordonnées sphériques (θ, ϕ) dont le vecteur normé associé est $(\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta)$.

Paramétrisation de la surface. Le patch de surface qui interagit avec la lumière est localement planaire et se trouve dans le plan de directions tangentes $X = (1, 0, 0)$ et $Y = (0, 1, 0)$. Sa normale est notée $\omega_g = (0, 0, 1)$ (où g signifie "géométrique").

Notations pour les directions. Les variables ω (avec ou sans index) dénotent toujours un vecteur normalisé. En plus de ω_g , la normale de la surface, nous notons ω_o la direction sortante (où l'index o signifie "outgoing") et ω_i la direction incidente (où l'index i signifie "incident"). Les quantités associées à ces directions seront toujours indexées de la même manière. Par exemple, pour la direction sortante ω_o , les coordonnées cartésiennes sont données par (x_o, y_o, z_o) et les coordonnées sphériques par (θ_o, ϕ_o) .

$\omega = (x, y, z)$	paramétrisation cartésienne
$\omega = (\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta)$	paramétrisation sphérique
$\omega_g = (0, 0, 1)$	normale géométrique
$\omega_o = (x_o, y_o, z_o)$	direction sortante
$\omega_i = (x_i, y_i, z_i)$	direction incidente

TABLE 2.2 – Notations : configuration géométrique.

2.3 Définition

Dans cette section, nous rappelons la définition formelle de la BRDF ainsi que son utilisation dans l'équation d'illumination locale. Les notations pour les opérateurs mathématiques que nous utilisons sont définies dans la table 2.3.

Définition formelle. La Bidirectional Reflectance Distribution Function (BRDF), est une fonction 4D (elle prend en argument deux directions normalisées) qui définit comment la lumière est réfléchiée par un patch de surface. Sa définition formelle date de 1965 et elle est due à Fred Nicodemus [Nic65], qui l'écrit de la manière suivante :

$$f(\omega_o, \omega_i) = \frac{\partial L(\omega_o)}{\partial E(\omega_i)} = \frac{\partial L(\omega_o)}{L(\omega_i) \cos \theta_i \partial \omega_i}. \quad (2.1)$$

La BRDF est ainsi définie comme étant le rapport entre la radiance sortante $L(\omega_o)$ et la radiance incidente $L(\omega_i)$ projetée sur le patch de surface (ou encore l'irradiance venant de la direction ω_i). L'équation est exprimée sous forme différentielle de manière à ce que seule la contribution de la direction ω_i soit prise en compte dans la BRDF. En effet, dans le rapport des quantités $\frac{L(\omega_o)}{L(\omega_i) \cos \theta_i}$, on ne sait pas quelle est la proportion dans $L(\omega_o)$ qui est réfléchi depuis la direction ω_i . Mettre la fraction sous forme différentielle permet de savoir exactement comment augmente $L(\omega_o)$ lorsque la radiance incidente augmente uniquement dans la direction ω_i . A noter que cette définition modélise une réponse linéaire de la surface : si la radiance incidente est multipliée par deux, on mesure deux fois plus de radiance sortante.

Propriétés. Une BRDF est dite *physiquement plausible* si elle vérifie les propriétés suivantes :

- Positivité : $f(\omega_o, \omega_i) \geq 0$. Cette propriété signifie simplement que l'énergie peut s'ajouter mais pas se soustraire : l'énergie négative n'existe pas.
- Réciprocité : $f(\omega_o, \omega_i) = f(\omega_i, \omega_o)$. De la même manière que lorsqu'on peut voir quelqu'un dans un miroir celui-ci peut nous voir aussi, la propriété de réciprocité vient du fait que si un chemin optique est possible dans un sens, alors il est aussi possible dans le sens inverse.
- Conservation de l'énergie : $\int_{\Omega} f(\omega_o, \omega_i) \cos \theta_i d\omega_i \leq 1$. La surface réfléchit la lumière incidente mais n'introduit pas d'énergie par elle-même : ce qui sort ne peut pas excéder ce qui rentre. Une question intéressante est de savoir si le bilan d'énergie peut être *exactement* 1. Nous y reviendrons dans le chapitre 4.

A noter qu'il y a une différence entre une BRDF physiquement plausible et une BRDF *physically based* (basée sur un modèle physique). Les premières satisfont les trois propriétés ci-dessus mais leur formulation peut être trouvée de manière empirique. Les secondes sont le résultat d'une modélisation physique. Lorsque la modélisation est bien définie, les BRDFs basées-physique sont physiquement plausibles. Nous y reviendrons dans le chapitre 4.

Equation d'illumination locale. La BRDF intervient dans l'équation d'illumination locale, historiquement nommée "du rendu", formalisée officiellement par Kajiyama [Kaj86] :

$$L(\omega_o) = \int_{\Omega} L(\omega_i) f(\omega_o, \omega_i) \langle \omega_i, \omega_g \rangle d\omega_i.$$

Elle signifie simplement que la radiance totale réfléchi par la surface est la radiance incidente multipliée par la BRDF depuis toutes les directions. L'objet d'un moteur de rendu est de calculer cette intégrale¹.

$\omega \cdot \omega$	produit scalaire
$ \omega \cdot \omega $	valeur absolue du produit scalaire
$\langle \omega, \omega \rangle$	produit scalaire clampé à 0
$\chi^+(a)$	fonction de Heaviside : 1 si $a > 0$ et 0 si $a \leq 0$

TABLE 2.3 – Notations : opérateurs mathématiques.

2.4 Modèles idéaux

Il existe deux modèles idéaux de surfaces réfléchissantes : le modèle parfaitement spéculaire et le modèle parfaitement diffus.

1. Dans le cas des surfaces réfléchissantes uniquement. Par exemple, les surfaces réfractantes, ou encore les volumes, sont modélisés par d'autres équations.

Réflexion spéculaire. Une surface spéculaire a le comportement optique d'un miroir parfait : la lumière est réfléchiée dans une seule direction, qu'on appelle la *direction réfléchiée* ω_r , et qui est donnée par l'équation :

$$\omega_r = -\omega_o + 2\omega_g |\omega_o \cdot \omega_g|. \quad (2.2)$$

La BRDF s'écrit alors :

$$f(\omega_o, \omega_i) = \frac{\delta_{\omega_r}(\omega_i)}{\cos \theta_i},$$

où δ est la distribution de dirac.

Réflexion diffuse. La loi de Lambert, qui date du 18^è siècle [Lam60], décrit les surfaces diffuses parfois aussi appelées surfaces *lambertiennes*. Un réflecteur parfaitement diffus émet une radiance qui ne dépend que de l'irradiance incidente. Autrement dit, la surface rediffuse uniformément dans toutes les directions la lumière incidente, ou, plus formellement, la BRDF est constante :

$$f(\omega_o, \omega_i) = \lambda, \quad (2.3)$$

où le facteur λ doit être plus petit que $\frac{1}{\pi}$ pour garantir la conservation de l'énergie.

2.5 Modèles empiriques

Les modèles empiriques de BRDF modélisent des effets observables de la réflectance des matériaux mais ne sont pas issus d'une modélisation physique. Ils ont souvent été introduits pour modéliser des surfaces *glossy* (brillantes) qui réfléchissent la lumière avec une dispersion autour d'une direction privilégiée, généralement la direction réfléchiée.

Le modèle de Phong. Le premier modèle empirique dans cet esprit est le modèle originel de Phong [Pho75], dont l'idée était simple : une manière de modéliser une dispersion autour de la direction réfléchiée est de remplacer le dirac δ_{ω_r} du modèle spéculaire par un lobe directionnel k_{ω_r} :

$$f(\omega_o, \omega_i) = k_{\omega_r}(\omega_i).$$

Cette manière de modéliser la BRDF introduit une propriété qui n'est pas observée sur des matériaux réels : la forme du lobe de la BRDF est invariante autour de la direction réfléchiée. En effet, les matériaux réels tendent à avoir un comportement anisotrope aux angles rasants. Cet effet est visible dans la figure 2.1 : le reflet du soleil sur la surface de la mer est très étiré : c'est parce que le lobe de la BRDF associée à la surface de la mer est anisotrope.

Le modèle de Blinn-Phong. Pour améliorer le modèle de Phong, Blinn et Phong proposèrent d'évaluer le lobe directionnel autour du *half-vector* plutôt qu'autour de la direction réfléchiée [Bli77] :

$$f(\omega_o, \omega_i) = k_{\omega_g}(\omega_h),$$

où ω_h est le half-vector est le vecteur normalisé à mi-chemin entre les deux directions :

$$\omega_h = \frac{\omega_o + \omega_i}{\|\omega_o + \omega_i\|}.$$

Cette manière d'écrire la BRDF produit le comportement anisotrope attendu. En effet, l'anisotropie vient essentiellement de la transformation entre le half-vector et le vecteur réfléchi. La préférence pour une paramétrisation dans l'espace des normales plutôt que dans l'espace des directions réfléchies est confirmée par d'autres études [FW94, Rus98, NDM05]. Il est important de noter que ce modèle se rapproche des modèles à micro-facettes spéculaires, dans le sens où le lobe k_{ω_g} joue le rôle de la distribution de normales de la micro-surface. En revanche, ce modèle n'est pas un modèle à micro-facettes, car il manque les termes de normalisation appropriés [Lew94].



FIGURE 2.1 – La réflexion glossy sur un matériau réel produit un effet d’étirement anisotrope aux angles rasant (gauche) qui n’est pas représenté avec des modèles qui supposent un lobe de BRDF constant autour de la direction réfléchie (droite).

Extensions anisotropes. Le modèle de Blinn-Phong sert de base à plusieurs modèles ultérieurs, en particulier certains qui introduisent des lobes de normales anisotropes [War92, AS00]. Ces nouveaux modèles se rapprochent encore plus des modèles à micro-facettes : ils sont correctement normalisés (ils conservent l’énergie) et comptent parfois un terme de Fresnel.

Modèles arbitraires. Le modèle de Lafortune [LFTG97] généralise le modèle originel de Phong en utilisant plusieurs lobes orientés par rapport à la direction de réflexion plutôt qu’un seul. De cette manière, la BRDF peut incorporer “en dur” divers phénomènes, comme l’anisotropie ou la rétro-réflexion.

Chapitre 3

Rappels sur la théorie des micro-facettes

Aujourd’hui, la théorie des micro-facettes est un des fondamentaux de l’image de synthèse. Par exemple, tous les ans à SIGGRAPH, le cours *physically based rendering* commence par une introduction à cette théorie [MHH⁺12, MHM⁺13]. Les modèles de BRDFs basés-physiques qui en découlent ont aujourd’hui pris le dessus sur les modèles empiriques à la fois dans le monde de la production pour les films mais aussi pour les applications temps-réel comme les jeux vidéos.

La théorie des micro-facettes modélise la BRDF émergente d’une *micro-surface statistique* faite d’un certain *modèle physique de matériau*. Elle fut développée originellement dans le domaine de l’optique physique pour étudier le comportement optique des surfaces rugueuses [BS63] et fut introduite dans la communauté de l’image de synthèse par les travaux de Torrance [TS67, CT82].

3.1 Le modèle statistique de micro-surface

Une instance de surface est décrite par ses propriétés spatiales : on sait que tel point de la surface a telles coordonnées, telle normale, telle visibilité pour telle direction etc. Mais il serait compliqué de faire une étude exhaustive de l’ensemble des surfaces possibles. La théorie des micro-facettes étudie les propriétés optiques d’une classe de surfaces qui partagent des propriétés statistiques, plutôt que sur une instance particulière. Les propriétés statistiques qui caractérisent la micro-surface sont *la distribution de normales D* et *la probabilité de masquage et d’ombrage G_2* .

3.1.1 La distribution de normales D

La distribution de normales D décrit la statistique des micro-normales ω_m dont la micro-surface est constituée. Le patch de surface macroscopique qui supporte la micro-surface est d’aire 1m^2 . L’aire de la micro-surface projetée sur la macro-surface doit avoir la même aire. La distribution D satisfait donc la propriété suivante :

$$\int_{\Omega} (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m = 1. \quad (3.1)$$

La définition de la distribution de normales est détaillée dans le chapitre 4.

Les modèles paramétriques. La distribution de normales historique est celle de Beckmann [BS63], qui correspond à une micro-surface à statistiques gaussiennes. La distribution de Laplace est souvent mentionnée dans la bibliographie de la propagation des ondes radio [Bro80, BBS02]. Le modèle de Trowbridge-Reitz [TR75], repris en image de synthèse par Walter et al. [WMLT07] et aujourd’hui appelé “GGX”, décrit

mieux les matériaux à distributions fractales. Bagher et al. [BSH12] proposent le modèle SGD dans le but de mieux approximer des matériaux mesurés.

La rugosité. Les modèles paramétriques prennent souvent un ou plusieurs paramètres de *rugosité*. Une surface parfaitement lisse n'a qu'une seule normale alignée avec la macro-normale : $D(\omega_m) = \delta_{\omega_g}(\omega_m)$. Plus la surface est rugueuse, plus la BRDF a un comportement glossy : la dispersion de la lumière augmente. Le paramètre de rugosité est souvent noté α où est parfois un couple (α_x, α_y) pour les BRDFs paramétriques anisotropes [RDP05]. Lorsque ces paramètres ont besoin d'être explicités dans les équations, nous noterons $D(\omega_m, \alpha)$ ou $D(\omega_m, \alpha_x, \alpha_y)$ à la place de $D(\omega_m)$.

L'espace des pentes. Certaines distributions de normales viennent de modèles où la micro-surface est un height field. C'est le cas par exemple de Beckmann, Laplace, GGX et SGD. En revanche, ce n'est pas le cas de la distribution de Phong. Lorsque la micro-surface est une height map, l'espace des micro-normales $\omega_m = (x_n, y_n, z_n)$ est en bijection avec l'espace des pentes $(x_{\tilde{m}}, y_{\tilde{m}})$ de la micro-surface. Les transformations pentes/normales sont données par :

$$(x_{\tilde{m}}, y_{\tilde{m}}) = \left(-\frac{x_m}{z_m}, -\frac{y_m}{z_m}\right),$$

et réciproquement :

$$\omega_m = \frac{(-x_{\tilde{m}}, -y_{\tilde{m}}, 1)}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}}.$$

On note P^{22} la distribution des pentes, dont la relation avec la distribution des normales est donnée par :

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) = \frac{D(\omega_m)}{\cos^4 \theta_m}.$$

La distribution des pentes décrit la proportion de pentes projetées sur la macro-surface et, pour la même raison que la distribution de normales, c'est cette projection qui est normalisée :

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dx_{\tilde{m}} dy_{\tilde{m}} = 1.$$

3.1.2 La fonction de masquage G_1

Pour qu'une micro-facette ω_m réfléchisse de la lumière de ω_i vers ω_o , il faut qu'elle soit visible depuis ces deux directions. La probabilité de visibilité dans une direction ω_o est donnée par la fonction de masquage $G_1(\omega_o, \omega_m)$. Le masquage pour la direction incidente ω_i est souvent appelé l'ombrage $G_1(\omega_i, \omega_m)$. Le masquage et l'ombrage sont indispensables pour garantir la conservation de l'énergie. En effet, si les contributions des micro-facettes masquées ou ombrées n'étaient pas soustraites de la BRDF, la micro-surface pourrait réfléchir plus d'énergie qu'elle n'en reçoit. Il existe de nombreuses variantes de cette fonction dans la littérature.

Le modèle de V-cavités. Le modèle introduit par Torrance [TS67, CT82] suppose que les micro-normales de la surface sont organisées de manière à former des cavités en forme de V de longueur infinie. Sous cette hypothèse, la proportion de chaque micro-facette visible dans les deux directions ω_o et ω_i peut-être dérivée analytiquement avec des calculs trigonométriques et vaut :

$$G_1(\omega_o, \omega_m) = \min \left(1, 2 \frac{(\omega_m \cdot \omega_g)(\omega_o \cdot \omega_g)}{\langle \omega_o, \omega_m \rangle} \right). \quad (3.2)$$

Ce modèle est souvent caractérisé de "peu réaliste", car il n'est pas possible qu'une infinité de V-cavités de longueur infinie cohabitent sur la même surface. Nous y reviendrons dans le chapitre 4. D'autres modèles sont des approximations de celui-ci, parfois plus facile à implémenter et moins cher à évaluer [KSK01, KSKK10].

Le modèle de Smith. La fonction de masquage de Smith fut originellement dérivée par Wagner et Smith [Wag67, Smi67] pour les surfaces à statistiques gaussiennes (distributions de Beckmann). Ce résultat a été généralisé plus tard à des types de distributions D arbitraires par Brown [Bro80] et a été rendu populaire récemment dans la communauté de l’image de synthèse par Walter et al. [WMLT07]. La forme générale de la fonction de masquage est donnée par :

$$G_1(\omega_o, \omega_m) = \frac{\chi^+(\omega_o \cdot \omega_m)}{1 + \Lambda(\omega_o)}, \quad (3.3)$$

où la fonction Λ dépend de la distribution des pentes P^{22} de la micro-surface. Elle est dérivée par une méthode qui modélise du *ray casting* sur la micro-surface statistique : elle calcule la probabilité qu’un rayon, partant d’un point de la micro-surface dans la direction d’observation, intersecte la micro-surface en un autre point. Elle est souvent décrite comme une “*approximation meilleure que celles du modèle à V-cavités*”. Nous y reviendrons dans le chapitre 4.

Le modèle implicite. Le modèle implicite est une approximation qui a pour but principal de simplifier l’implémentation de la BRDF [MHM⁺13]. Il est essentiellement basé sur l’observation suivante : en vue frontale ($\omega_o \cdot \omega_g = 1$), aucune partie de la surface n’est masquée et la probabilité de masquage vaut 1. En vue rasante ($\omega_o \cdot \omega_g = 0$), presque toute la surface est masquée et la probabilité de masquage vaut 0. On constate que dans les deux cas extrêmes la probabilité de masquage prend la même valeur que le produit scalaire : $\omega_o \cdot \omega_g$. Le modèle implicite pose arbitrairement la fonction de masquage comme étant ce produit scalaire et de la même manière la fonction d’ombrage est donnée par $\omega_i \cdot \omega_g$. Comme ces deux produits scalaires sont présents au dénominateur de la BRDF, son expression se simplifie et devient :

$$f(\omega_o, \omega_i) = \frac{F(\omega_i, \omega_h) D(\omega_h)}{4}.$$

Le modèle basé sur la conservation de la surface projetée. Ashikhmin et al. ont observé que l’*aire projetée de la surface visible* dans n’importe quelle direction est une quantité qui doit être la même pour la macro-surface et la micro-surface [APS00]. En se basant sur cette propriété et en supposant que la probabilité de masquage est décorrélée de l’orientation des normales, il est possible de dériver une fonction de masquage pour n’importe quelle distribution de normales :

$$G_1(\omega_o, \omega_m) = \chi^+(\omega_o \cdot \omega_m) \frac{\cos \theta_o}{\int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m}. \quad (3.4)$$

Cette fonction n’a pas d’expression analytique dans le cas général et, en pratique, ils la pré-calculent et la stockent dans une table. Cependant, dans le chapitre 4, nous verrons que les hypothèses du modèle d’Ashikhmin sont exactement les mêmes que celles du modèle de Smith. D’ailleurs, Smith [Smi67] note que sa fonction de masquage satisfait la propriété de la conservation de l’aire projetée. Certains des résultats de l’article Ashikhmin et al. présentent des BRDFs calculées avec des micro-surfaces à statistiques gaussiennes. Pour ces résultats, sans en avoir conscience, la fonction de masquage qu’ils pré-calculent et stockent n’est autre que la fonction de Smith pour laquelle une forme analytique est disponible.

3.1.3 La fonction de masquage et d’ombrage G_2

Pour qu’une micro-facette ω_m réfléchisse de la lumière de ω_i vers ω_o , il faut qu’elle soit visible *simultanément* depuis ces deux directions. C’est cette probabilité jointe qui est représentée par la fonction de masquage et d’ombrage $G_2(\omega_o, \omega_i, \omega_m)$. Walter et al. énoncent les propriétés qu’elle doit satisfaire pour être physiquement plausible [WMLT07] :

- C’est une probabilité, donc à valeurs dans $[0, 1]$: $0 \leq G_2(\omega_o, \omega_i, \omega_m) \leq 1$
- Elle est symétrique : $G_2(\omega_o, \omega_i, \omega_m) = G_2(\omega_i, \omega_o, \omega_m)$

- Les normales auto-masquées ou auto-ombrées ne sont pas visibles :
 $G_2(\omega_o, \omega_i, \omega_m) = 0$ si $(\omega_o \cdot \omega_m)(\omega_o \cdot \omega_g) \leq 0$ ou $(\omega_i \cdot \omega_m)(\omega_i \cdot \omega_g) \leq 0$

Il existe différentes formes qui tiennent compte de différents types de corrélations entre le masquage et l'ombrage.

Masquage et ombrage séparables. La forme la plus simple est la forme séparable dans laquelle les événements de masquage et d'ombrage sont supposés indépendants et sont multipliés :

$$G_2(\omega_o, \omega_i, \omega_m) = G_1(\omega_o, \omega_m) G_1(\omega_i, \omega_m). \quad (3.5)$$

Cette forme ne tient pas compte des corrélations possibles entre le masquage et l'ombrage : elle produit toujours une surestimation, c'est un majorant de la valeur exacte.

Masquage et ombrage corrélés par les hauteurs. Lorsque la micro-surface est un height field, la probabilité qu'une micro-facette soit visible est corrélée avec sa hauteur : plus elle est élevée, plus la probabilité d'être visible est importante. De fait, la corrélation par les hauteurs introduit une corrélation entre le masquage et l'ombrage.

Dans le modèle à V-cavités, le masquage et l'ombrage donnent la proportion supérieure de la micro-facette visible. La partie visible simultanément s'arrête donc au minimum des deux et s'écrit [TS67, CT82] :

$$G_2(\omega_o, \omega_i, \omega_m) = \min(G_1(\omega_o, \omega_m) G_1(\omega_i, \omega_m)). \quad (3.6)$$

Bourlier et al. ont dérivé le modèle de Smith avec corrélation par les hauteurs [BBS02, BB03]. La corrélation est introduite de la manière suivante :

$$G_2(\omega_o, \omega_i, \omega_m) = \frac{\chi^+(\omega_o \cdot \omega_m) \chi^+(\omega_o \cdot \omega_m)}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)}. \quad (3.7)$$

La dérivation de cette fonction est donnée dans l'annexe A.2.

Masquage et ombrage corrélés azimuthalement. Le masquage et l'ombrage sont aussi corrélés lorsque les directions ω_i et ω_o sont proches. Typiquement, lorsque $\omega_o = \omega_i$, ils sont totalement corrélés puisque les micro-facettes visibles pour ω_o le sont aussi pour ω_i et réciproquement. Dans ce cas extrême, la fonction de masquage et d'ombrage est simplement la fonction de masquage (ou d'ombrage) :

$$G_2(\omega_o, \omega_i, \omega_m) = G_1(\omega_o, \omega_m) = G_1(\omega_i, \omega_m).$$

cet effet s'appelle le *hotspot* : lorsque la direction d'observation et la direction de la lumière sont parallèles, les ombres disparaissent. Cela ne signifie pas qu'il n'y a pas d'ombre sur la surface, mais plutôt qu'il n'y a pas d'ombre visible. Comme la BRDF mesure la radiance mesurée dans la direction d'observation, si les ombres ne sont pas visibles depuis cette direction, elles ne doivent pas être prises en compte dans la BRDF.

Dans le cas général, le masquage et l'ombrage ne sont ni totalement corrélés ni totalement décorrés. Ashikhmin et al. incorporent cette corrélation en interpolant ces deux cas extrêmes [APS00] :

$$G_2(\omega_o, \omega_i, \omega_m) = (1 - r(\phi)) G_1(\omega_o, \omega_m) G_1(\omega_i, \omega_m) + r(\phi) \min(G_1(\omega_o, \omega_m) G_1(\omega_i, \omega_m)),$$

où la fonction $r(\phi)$ est un facteur empirique similaire à celui de Ginneken et al. qu'on présente ci-dessous.

Masquage et ombrage corrélés par les hauteurs et azimuthalement. Le modèle de Smith avec corrélation sur les hauteurs peut être enrichi en introduisant un facteur de corrélation azimuthale :

$$G_2(\omega_o, \omega_i, \omega_m) = \frac{\chi^+(\omega_o \cdot \omega_m) \chi^+(\omega_o \cdot \omega_m)}{1 + \max(\Lambda(\omega_o), \Lambda(\omega_i)) + (1 - r(\omega_o, \omega_i)) \min(\Lambda(\omega_o), \Lambda(\omega_i))}. \quad (3.8)$$

Sous cette forme, le masquage et l’ombrage sont totalement corrélés ($r = 0$) lorsque $\omega_o = \omega_i$. La corrélation décroît lorsque l’angle entre les deux directions augmente. Lorsque les deux sont totalement décorrélés ($r = 1$), on retrouve la forme qui tient compte des corrélations sur les hauteurs uniquement.

Ginneken et al. proposent un facteur empirique $1 - r = \frac{4.41\phi}{4.41\phi + 1}$ qui dépend de ϕ , la différence d’angle azimuthal entre ω_o et ω_i [vGSK98]. La limitation de cette heuristique est qu’elle ne tient pas compte de la rugosité de la surface ni des angles d’élévation θ_o et θ_i .

Dans le chapitre 6, nous proposons une modélisation de cette corrélation et une dérivation du facteur r .

3.2 Les modèles physiques de matériaux

Le modèle de matériau décrit l’interaction entre les micro-facettes et la lumière. Choisir un matériau revient à choisir la micro-BRDF $\rho_{\mathcal{M}}$ associée aux micro-facettes. La macro-BRDF ρ qu’on cherche à modéliser étant la résultante, ou l’effet émergent, de l’ensemble de ces interactions microscopiques.

3.2.1 Les matériaux diffractants

L’équation de diffraction de Kirchhoff modélise la propagation d’une onde électromagnétique interagissant avec une surface [BS63]. Cette équation est utilisée notamment lorsque les micro-facettes sont d’une taille comparable, ou sont plus petites, que la longueur d’onde de la lumière. Dans ce cas, le modèle de la géométrie optique n’est pas valide. He et al. [HTSG91] proposent un modèle de BRDF basé sur l’optique ondulatoire. Stam [Sta99] utilise l’équation de Kirchhoff pour calculer des shaders de matériaux diffractants.

3.2.2 Le modèle à micro-facettes spéculaires

Le modèle à micro-facettes spéculaires est un cas particulier de l’équation de Kirchhoff qui s’applique lorsque les micro-facettes sont d’une taille significativement plus grande que la longueur d’onde de la lumière [Sta99], c’est-à-dire lorsque seuls des effets de géométrie optique interviennent. On peut le dériver en partant de l’équation de Kirchhoff ou par une approche entièrement basée sur la géométrie optique [TS67, CT82]. Nous proposons une telle dérivation dans le chapitre 4.

L’équation de Fresnel. Lorsque les micro-facettes sont des miroirs parfaits qui obéissent aux lois de Snell-Descartes [PH10] : la lumière incidente est réfléchiée et réfractée. La micro-BRDF, qui ne rend compte que de l’énergie réfléchiée (l’énergie réfractée étant modélisée par la BTDF), correspond au modèle spéculaire idéal donné par l’équation (2.2). La répartition d’énergie réfléchiée et réfractée dépend de la nature du matériau (conducteur ou diélectrique) et de son indice de réfraction et elle est donnée par les équations de Fresnel. Ces équations sont complexes et leur évaluation est coûteuse. Pour simplifier, Schlick [Sch94] propose une approximation, en supposant une polarisation moyennée :

$$F(\cos \theta = \omega_i \cdot \omega_h) \approx F(0) + (1 - F(0))(1 - \cos \theta)^5,$$

où $F(0)$ est la valeur de la fonction de Fresnel en incidence frontale ($\omega_i = \omega_h$). Cette approximation est aujourd’hui très largement utilisée en image de synthèse.

La macro-BRDF. Le modèle de BRDF à micro-facettes spéculaires est donné par l’équation :

$$\rho(\omega_o, \omega_i) = \frac{F(\omega_i, \omega_h) G_2(\omega_i, \omega_o, \omega_h) D(\omega_h)}{4 |\omega_i \cdot \omega_g| |\omega_o \cdot \omega_g|}. \quad (3.9)$$

On voit que la distribution de normales D est évaluée sur le half-vector ω_h . En effet, puisque les micro-facettes sont spéculaires, seules les micro-facettes orientées dans la même direction que le half-vector ω_h réfléchissent de la lumière depuis la direction incidente ω_i vers la direction sortante ω_o . Le terme de Fresnel donne la proportion d’énergie réfléchiée par les micro-facettes de normale ω_h et le terme de masquage et d’ombrage G_2 modélise le fait que seules les micro-facettes visibles dans les deux directions simultanément contribuent effectivement à la BRDF.

3.2.3 Le modèle à micro-facettes diffuses

Le modèle à micro-facettes diffuses [ON94] est proche du modèle à micro-facettes spéculaires, la différence étant que les micro-facettes ne sont plus modélisées comme des miroirs parfaits, mais comme un matériau idéal parfaitement diffus, dont nous avons donné la BRDF en équation (2.3).

Si l'équation (3.9) du modèle à micro-facettes spéculaires a une forme analytique, c'est parce que seules les micro-facettes orientées parallèlement à la direction du half-vector contribuent à la BRDF. Lorsque les micro-facettes sont diffuses, elles contribuent toutes en même temps et l'équation ne se simplifie pas de cette manière. Dans le cas général, elle n'a pas de forme analytique. Oren et Nayar proposent une approximation analytique lorsque la distribution de normale est un lobe gaussien isotrope avec le modèle de masquage des V-cavités [ON94]. Un autre modèle est dérivé par van Ginneken et al. avec une distribution de Beckmann isotrope et avec le modèle de masquage de Smith [vGSK98].

3.2.4 Les modèles multi-couches

Certains modèles supposent que la micro-surface n'est pas faite d'un seul matériau, mais plutôt de plusieurs couches de différents matériaux. Les interactions entre les différentes couches sont souvent complexes [WW07, Sim09]. Les modèles les plus simples mettent en jeu une couche spéculaire posée sur une surface diffuse. Dans ce cas, c'est le terme de Fresnel qui contrôle la quantité d'énergie répartie entre la partie spéculaire et la partie diffuse [Sch94, KSK01], contrairement au modèle de Phong où elles sont simplement multipliées par des facteurs constants et additionnées. On retrouve ce phénomène dans les peintures avec vernis, la peau (graisse, pigments) ou encore des matériaux mouillés.

Chapitre 4

Comprendre la fonction de masquage et d’ombrage dans les modèles à micro-facettes



Matt Pharr @mattpharr · 16 avr.

A lot of good intuitions presented, well-thought out figures, etc. I've learned a lot!

Ouvrir

↳ Répondre ↻ Retweeter ★ Favori ⋮ Plus



Matt Pharr @mattpharr · 16 avr.

I've been reading Heitz's TR on microfacets hal.inria.fr/hal-00942452; it's incredible. Interesting results, explained remarkably clearly.

Ouvrir

↳ Répondre ↻ Retweeter ★ Favori ⋮ Plus

Matt Pharr sur Twitter, 16 avril 2014.

La théorie des micro-facettes est considérée comme une discipline mature, ses premiers résultats datant des années 60. Pourtant, nous verrons que la littérature sur le sujet laisse de nombreuses questions en suspens. Mais les réponses à ces questions sont indispensables pour partir sur des bases saines. En effet, l’objectif de la partie I de cette thèse est d’étudier jusqu’à quel point le modèle des micro-facettes peut être utilisé ou étendu pour représenter des surfaces multi-échelles. Mais il est primordial, avant de chercher à l’étendre, de comprendre ce qu’il modélise exactement, ses propriétés, ses limitations, etc. C’est l’objet de ce chapitre. Dans le but de nous approprier la théorie, nous y redécrivons l’essentiel des résultats déjà connus sur les BRDFs à micro-facettes en explicitant, à chaque étape, toutes les hypothèses que nous utilisons.

Si de nombreux résultats dérivés dans ce chapitre étaient déjà connus dans la littérature, notre contribution majeure est de rassembler les pièces du puzzle et de proposer une vision d’ensemble et des intuitions originales de cette théorie. L’intuition la plus importante obtenue dans ce chapitre est probablement que la quantité fondamentale dans la théorie des micro-facettes n’est pas, contrairement à ce que les gens ont tendance à penser, la distribution de normales, mais plutôt la distribution de normales *visibles*. Cette distribution est le point de départ pour dériver tous les résultats du modèle. Cette intuition a aussi des implications pratiques importantes. Par exemple, c’est l’idée principale derrière la stratégie d’importance sampling proposée dans le chapitre 11 ou la technique de filtrage des matériaux multi-échelles dans le chapitre 9.

Au coeur de nos dérivations se trouve la fonction de masquage dont le rôle est souvent sous-estimé et incompris dans la littérature. Nous montrons que la fonction de masquage sert à normaliser la distribution de normales visibles et, en conséquence, garantit la conservation *exacte* de l’énergie dans les BRDFs. D’ailleurs, nous verrons que l’usage même du mot “exact” n’est pas clair dans la littérature en ce qui concerne la fonction de masquage. Qu’est-ce qu’une fonction de masquage exacte ? La réponse se trouve dans ce chapitre.

Nos dérivations nous mènent aussi à certains résultats originaux. Nous proposons le *Weak White Furnace Test*, lié à la conservation de l’énergie de la BRDF, comme critère strict de validation d’un modèle à micro-facettes. Nous l’utiliserons dans le chapitre 5 pour valider les équations du modèle multi-échelles que nous y proposons. Un autre résultat original que nous dérivons est l’invariance par étirement de la fonction de masquage. Cette propriété montre que la fonction de masquage a une forme close sur certaines micro-surfaces anisotropes, rend sa dérivation triviale et permet son utilisation en pratique dans des modèles de BRDFs.

4.1 Questions ouvertes dans la littérature

4.1.1 Sur le choix de la fonction de masquage

Nous avons vu dans le chapitre 3 que les BRDFs à micro-facettes sont constituées de plusieurs composants, dont chacun a plusieurs variantes, le tout offrant une combinatoire importante de choix possibles pour la mise au point d'un modèle. Choisir les ingrédients d'une BRDF à micro-facettes n'est pas toujours évident, en particulier pour la fonction de masquage et d'ombrage. En effet, la nature de l'approximation de Schlick pour le terme de Fresnel est explicite (les équations exactes du modèle étant connues) et il existe assez peu d'alternatives de toute manière. En ce qui concerne la distribution de normales D , il est assez intuitif de voir comment elle influence la forme de la BRDF. En revanche, si nous savons que le rôle de la fonction de masquage et d'ombrage est de garantir la conservation de l'énergie, les raisons et les implications de ce choix sont moins transparentes.

Dans cette section, nous proposons de regarder plus en détail les argumentaires et les points de vue typiques qu'on trouve dans la littérature. Nous avons décidé de nous appuyer notamment sur des citations de *A Microfacet-based BRDF Generator* [APS00] et *Microfacet Models for Refraction through Rough Surfaces* [WMLT07], deux articles de référence pour les gens qui travaillent sur le sujet, et sur le cours SIGGRAPH de l'année 2013 [MHM⁺13] qui propose un état de l'art exhaustif sur la question.

Un choix arbitraire. La première chose qu'on observe en se documentant sur la fonction de masquage, c'est que son choix semble être arbitraire :

"Because there are many possible surface geometries that are consistent with a given $p(h)$, it is the case that no specific shadowing function is "right"."

Ashikhmin et al. [APS00]

"We use an approximation due to Smith, which was originally derived for gaussian surfaces and later generalized for arbitrary microfacet distributions."

Walter et al. [WMLT07]

"While choosing the family of Smith functions (as opposed to e.g. the Cook-Torrance geometry function) is a decision independent of the NDF used, once this choice is made you should use the correct Smith function for the NDF."

SIGGRAPH course notes 2013 [MHM⁺13]

Des modèles approximatifs. Si le choix est arbitraire, y a-t-il des critères qui permettent de le faire de manière éclairée? Les différentes fonctions de masquage sont souvent qualifiées d'"approximatives" ou encore de "simplifiées" :

"A rigorous derivation of the probability that a point on the surface is both visible and illuminated (also known as the bistatic shadowing function) leads to very complicated expressions and a set of approximations is made to make the problem tractable. Several forms of the shadowing term have been derived in different fields and some of them (usually after further simplification) were later introduced to computer graphics reflection models."

Ashikhmin et al. [APS00]

” Many approximations for the shadowing-masking term have been proposed. ”
Walter et al. [WMLT07]

” The shadowing-masking function depends on the details of the microsurface, and exact expressions are rarely available. More typically, approximations are derived using various statistical models and simplifying assumptions. ”
Walter et al. [WMLT07]

Mais concrètement, quelles sont ces approximations ? L’usage du mot “approximation” est ambigu : ce sont des approximations par rapport à quoi au juste ? Est-ce que ce sont des approximations du modèle physique ou des dérivations mathématiques des équations du modèle ? Il est toujours possible de comparer les modèles à des données mesurées, mais quel serait un modèle de référence exact ? Quelle est l’information perdue dans les modèles “simplifiés” actuels ?

Des descriptions qualitatives. Le lecteur reste sur sa faim lorsqu’il cherche des réponses à ces questions, les éléments de réponses étant souvent très qualitatifs : certains modèles de surface sont qualifiés de peu “plausibles” ou encore peu “réalistes” :

” The shadowing function by Torrance and Sparrow is simple, but assumes an inconsistent model of an isotropic surface exclusively made by very long V-cavities. ”
Ashikhmin et al. [APS00]

” The Cook-Torrance geometry function has been used a lot over the years, but it has some problems : it’s based on an unrealistic microgeometry model (an isotropic surface composed of infinitely long grooves) and it’s also unaffected by roughness. ”
SIGGRAPH course notes 2013 [MHM+13]

Mais que signifie exactement le mot “réaliste” dans ce contexte ? Quelle sont les conséquences pour la BRDF ?

4.1.2 Trouver des critères quantitatifs

Pour mieux comprendre et choisir un modèle à micro-facettes, il est nécessaire de trouver des critères quantitatifs qui le caractérisent. Nous en avons listé certains en section 3.1.3, mais ils ne sont pas assez discriminants, car la plupart des modèles existants satisfont déjà ces propriétés. Y a-t-il d’autres propriétés à exploiter pour mieux caractériser un modèle ?

La conservation de l’énergie. La fonction de masquage et d’ombrage a pour but de garantir la conservation de l’énergie. La propriété de conservation d’énergie est satisfaite lorsque la valeur de la BRDF est inférieure ou égale à 1 : $\int_{\Omega} f(\omega_o, \omega_i) \langle \omega_i, \omega_g \rangle d\omega_i \leq 1$.

Mais pourquoi pas *exactement* 1 ? Une surface qui n’absorbe pas et ne transmet pas la lumière (lorsque le terme de Fresnel vaut 1) ne devrait-elle pas réfléchir 100% de l’énergie incidente ? Pourquoi les modèles ne s’intègrent-ils pas *exactement* à 1 dans ce cas ? Où est passée l’énergie manquante ? Peut-on imposer l’intégration de la BRDF à exactement 1 comme un des critères de validation du modèle ?

La conservation de l’aire projetée. A la fin de son article, Smith [Smi67] note que la conservation de l’aire projetée de la surface visible est une propriété importante satisfaite par sa fonction de masquage, mais sans expliquer “pourquoi ça marche”. Il n’y a, à notre connaissance, aucune référence qui aborde cette question en détail, à l’exception du travail de Ross et al. [RDP05] qui l’effleurent dans leur étude de la

BRDF de la surface de la mer : ils modélisent la surface de la mer comme une surface gaussienne et dérivent le coefficient de normalisation de cette BRDF. Ils constatent que ce coefficient a la même forme que la fonction de masquage de Smith et que les deux se simplifient. Cette propriété est commode pour simplifier les calculs mais elle n’est pas plus discutée dans l’article. Est-ce un pur hasard ou bien y a-t-il quelque chose à comprendre ?

Ashikhmin et al. [APS00] utilisent directement cette propriété pour déduire et pré-calculer leur propre fonction de masquage. Cette propriété, qui semble pertinente, ne fait pourtant pas partie de la liste de Walter et al. [WMLT07] que nous avons rappelée en section 3.1.3. Pourquoi ce critère n’a-t-il pas rejoint les autres ? Toutes les fonctions classiques de masquage le satisfont-elles ? Si non, cela signifie-t-il qu’elles ne sont pas valides ? Ce critère peut-il être utilisé pour valider le modèle ?

Notre démarche. Dans le but de répondre à ces questions, nous refaisons une dérivation complète du modèle à micro-facettes en nous basant entièrement sur des propriétés de géométrie optique. Notre but est de pousser la dérivation le plus loin possible en explicitant toutes les hypothèses ou simplifications introduites dans le modèle. C’est de cette manière que nous pourrions comprendre ses propriétés, les conséquences des différentes options proposées dans la littérature et trouver les réponses aux questions que nous avons posées. Ces réponses sont données de manière succincte dans les sections “résumé” 4.2.5, 4.3.6 et 4.4.4.

Dans la section 4.2, nous partons de la propriété suivante : la fonction de masquage doit toujours satisfaire la conservation de l’aire projetée de la micro-surface visible. L’originalité de notre approche est de partir directement de cette propriété pour dériver l’ensemble des résultats du modèle de manière unifiée, dans le même cadre théorique, plutôt que d’utiliser des formulations ou des dérivations différentes pour chaque nouveau modèle.

Dans la section 4.3, nous définissons la distribution de normales visibles et montrons que la fonction de masquage sert à garantir sa normalisation. Nous montrons comment partir de cette distribution permet de redériver les équations des BRDFs à micro-facettes spéculaires ou diffuses. Nous en déduisons un critère strict de validation, que nous appelons le *Weak White Furnace Test*, qui mesure la conservation d’énergie de la BRDF en vérifiant qu’elle s’intègre à exactement 1 (et non simplement moins que 1).

Dans la section 4.4, nous montrons comment le choix d’un profil de micro-surface permet d’instancier la fonction de masquage. Nous donnons en exemple les profils associés au modèle de Smith et à celui en V-cavités. Nous montrons qu’avec un modèle de surface non corrélée, le modèle d’Ashikhmin et al. [APS00], nous retrouvons le modèle de Smith généralisé. C’est le résultat qu’avaient trouvé indirectement Ross et al. [RDP05]. Ainsi, par cette dérivation, nous unifions ces deux résultats qui étaient supposés être différents jusqu’à aujourd’hui. Nous passons aussi en revue d’autres fonctions de masquage qui ne correspondent à aucun modèle de micro-surface.

Enfin, dans la section 4.5, nous montrons comment des propriétés d’invariances topologiques de la configuration peuvent être utilisées pour étendre les modèles à des micro-surfaces anisotropes.

Ω	domaine sphérique (4π steradians)
(θ, ϕ)	coordonnées sphériques : $\omega = (\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta)$
$\omega_o = (x_o, y_o, z_o)$	direction sortante
$\omega_i = (x_i, y_i, z_i)$	direction incidente
$\omega_h = \frac{\omega_o + \omega_i}{\ \omega_o + \omega_i\ }$	half vector
\mathcal{G}	surface géométrique
$\omega_g = (0, 0, 1)$	normale géométrique
\mathcal{M}	micro-surface
p_m	point de la micro-surface
$\omega_m = (x_m, y_m, z_m)$	normale de la micro-surface
$\omega_1 \cdot \omega_2$	produit scalaire
$ \omega_1 \cdot \omega_2 $	valeur absolue du produit scalaire
$\langle \omega_1, \omega_2 \rangle$	produit scalaire clampé : 0 if $\omega_1 \cdot \omega_2 < 0$
$\chi^+(a)$	fonction de Heaviside : 1 si $a > 0$ et 0 si $a \leq 0$
$\delta(a)$	distribution de Dirac : $\int_{-\infty}^{+\infty} \delta(a) da = 1$

TABLE 4.1 – Notations mathématiques.

$L(\omega_o)$	radiance dans la direction ω_o	W/sr/m ²
$L(\omega_o, p_m)$	radiance dans la direction ω_o au point p_m	W/sr/m ²
$L(\omega_o, \mathcal{M})$	radiance dans la direction ω_o sur la micro-surface \mathcal{M}	W/sr/m ²
$\rho(\omega_o, \omega_i)$	BRDF	1/sr
$\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m)$	micro-BRDF de la normale ω_m	1/sr
$D(\omega_m)$	distribution de normales	m ² /sr
$D_{\omega_o}(\omega_m)$	distribution de normales visibles depuis ω_o	1/sr
$G_1(\omega_o, p_m)$	fonction de masquage spatiale au point p_m (binary value)	-
$G_1(\omega_i, p_m)$	fonction d'ombrage spatiale au point p_m (binary value)	-
$G_2(\omega_o, \omega_i, p_m)$	fonction de masquage et d'ombrage spatiale au point p_m (binary value)	-
$G_1(\omega_o, \omega_m)$	fonction de masquage statistique de la normale ω_m (in $[0, 1]$)	-
$G_1(\omega_i, \omega_m)$	fonction d'ombrage statistique de la normale ω_m (in $[0, 1]$)	-
$G_2(\omega_o, \omega_i, \omega_m)$	fonction de masquage et d'ombrage statistique de la normale ω_m (in $[0, 1]$)	-

TABLE 4.2 – Quantités physiques et les unités associées ('-' pour les quantités adimensionnelles).

4.2 Dérivation de la fonction de masquage

Dans cette section, nous rappelons comment la conservation de l'aire projetée de la micro-surface peut être utilisée pour contraindre la fonction de masquage [APS00]. Nous commençons par définir le concept d'aire projetée (4.2.1) et montrons en quoi il est essentiel à la définition physique de la radiance. Dans un second temps, nous définissons les équations statistiques de la théorie des micro-facettes (4.2.2). La conservation de la surface projetée (4.2.3) donne une nouvelle équation que nous utilisons pour contraindre la fonction de masquage (4.2.4). Cette contrainte, associée au choix d'un profil de micro-surface, mène à la dérivation de la fonction de masquage exacte.

4.2.1 Mesure de la radiance sur une surface

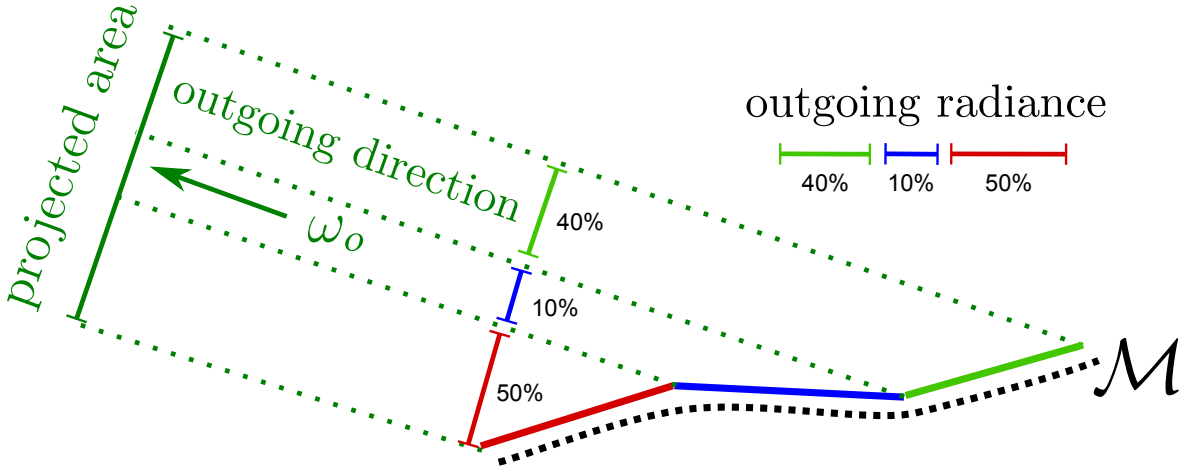


FIGURE 4.1 – La radiance sortant d'une surface est la moyenne de la radiance sur l'ensemble des points de la surface, pondérés par leurs aires projetées dans la direction sortante.

La radiance L est la densité d'énergie passant à travers une surface par angle solide et elle est mesurée en watts par stéradian par mètre carré ($\text{W}\cdot\text{sr}^{-1}\cdot\text{m}^{-2}$). La radiance $L(\omega_o, \mathcal{M})$ sortante d'une surface \mathcal{M} dans la direction ω_o , est la somme des radiances $L(\omega_o, p_m)$ sortant de chaque point p_m de la surface dans la direction ω_o et pondérées par les surfaces projetées de ces points dans la direction ω_o , comme illustré en figure 4.1 :

$$L(\omega_o, \mathcal{M}) = \frac{\int_{\mathcal{M}} \text{projected area}(p_m) L(\omega_o, p_m) dp_m}{\int_{\mathcal{M}} \text{projected area}(p_m) dp_m}. \quad (4.1)$$

L'aire projetée de chaque point dans la direction sortante est un facteur de pondération dont la somme $\int_{\mathcal{M}} \text{projected area}(p_m) dp_m$, est le coefficient de normalisation des fractions d'aires projetées. Diviser par cette somme est essentiel pour que les unités physiques soient correctes et assure que la densité d'énergie est préservée.

Dans les sections suivantes, nous allons voir, conformément à la théorie des micro-facettes, que les micro-facettes sont aussi pondérées par leurs aires projetées et que la fonction de masquage est le coefficient de normalisation requis pour la conservation de l'énergie.

4.2.2 Statistiques des micro-facettes

Nous considérons une région planaire que nous appelons la *surface géométrique* \mathcal{G} , dont l'aire vaut 1m^2 par convention : $\int_{\mathcal{G}} dp_g = 1\text{m}^2$. Le modèle des micro-facettes imagine qu'au niveau microscopique la surface n'est pas parfaitement plate mais est faite de micro-facettes décalées de la surface géométrique, dont l'ensemble forme ce qu'on appelle la *micro-surface* \mathcal{M} . Plus précisément, si ω_g est la normale de la surface géométrique \mathcal{G} , alors \mathcal{M} est l'ensemble des points de la micro-surface qui se projettent sur \mathcal{G} le long de ω_g . Chaque point p_m de la micro-surface \mathcal{M} a une normale $\omega_m(p_m)$.

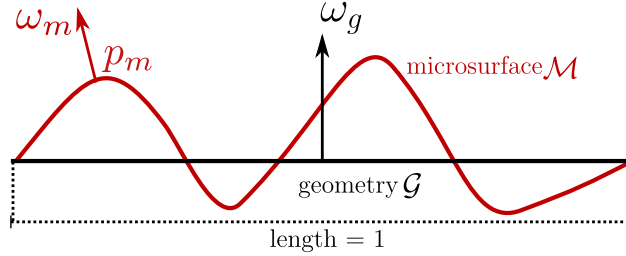


FIGURE 4.2 – La surface géométrique et la micro-surface.

La théorie des micro-facettes est un modèle statistique des propriétés optiques de la micro-surface. Donc, écrire des équations statistiques plutôt que spatiales est plus commode pour définir ce modèle. Ces statistiques sont souvent définies dans l'espace des normales, qui est le domaine sphérique Ω .

La distribution de normales. Pour faire le lien entre les intégrales définies sur la micro-surface et celles définies sur la sphère, c'est-à-dire convertir les équations spatiales en équations statistiques, nous avons besoin d'une mesure pour tenir compte des changements de domaines. La *distribution de normales* donne cette mesure. Elle se mesure en mètre carré par stéradian¹ ($\text{m}^2.\text{sr}^{-1}$) et elle est définie par :

$$D(\omega) = \int_{\mathcal{M}} \delta_{\omega}(\omega_m(p_m)) dp_m,$$

où les unités de la distribution de Dirac sont les inverses de celles de son argument (sr^{-1}). Considérons une région $\Omega' \subset \Omega$ de la sphère ainsi qu'un sous-ensemble $\mathcal{M}' \subset \mathcal{M}$ de la micro-surface, qui contient tous les points $p_m \in \mathcal{M}$ dont les normales $\omega_m(p_m)$ sont dans Ω' , tels que :

$$p_m \in \mathcal{M}' \iff \omega_m(p_m) \in \Omega'.$$

La distribution de normales est telle que son intégrale, sur n'importe quelle région $\Omega' \subset \Omega$ de la sphère, donne l'aire de l'ensemble $\mathcal{M}' \subset \mathcal{M}$ dont les normales sont dans Ω' :

$$\int_{\mathcal{M}'} dp_m = \int_{\Omega'} D(\omega_m) d\omega_m.$$

En conséquence, l'intégrale de la distribution de normales est l'aire de la micro-surface :

$$\text{microsurface area} = \int_{\mathcal{M}} dp_m = \int_{\Omega} D(\omega_m) d\omega_m.$$

1. Pour être parfaitement rigoureux, il faudrait l'appeler la distribution de normales *par mètre carré de surface géométrique* et c'est la raison pour laquelle nous la mesurons en ($\text{m}^2.\text{sr}^{-1}$). Walter et al. [WMLT07] la définissent d'une autre manière et la mesurent en sr^{-1} en introduisant les m^2 via une constante multiplicative externe à la distribution. Ce choix est une simple question de convention et n'impacte pas le reste des équations ni leur analyse dimensionnelle.

Equations spatiales et statistiques. Si $f(\omega_m)$ est une fonction des normales de la surface, l'intégration spatiale de f peut être remplacée par une intégration statistique :

$$\int_{\mathcal{M}} f(\omega_m(p_m)) dp_m = \int_{\Omega} f(\omega_m) D(\omega_m) d\omega_m,$$

où la partie de gauche de l'équation est l'*intégrale spatiale* et la partie de droite est l'*intégrale statistique*. Dans la figure 4.3(a), nous utilisons cette propriété où f est le produit scalaire.

Fonctions statistiques. Si $g(p_m)$ est une fonction spatiale définie sur la micro-surface, on définit la fonction statistique $g(\omega_m)$:

$$g(\omega) = \frac{\int_{\mathcal{M}} \delta_{\omega}(\omega_m(p_m)) g(p_m) dp_m}{\int_{\mathcal{M}} \delta_{\omega}(\omega_m(p_m)) dp_m}.$$

Les fonctions statistiques peuvent être utilisées dans des intégrales statistiques de la manière suivante :

$$\int_{\mathcal{M}} g(p_m) dp_m = \int_{\Omega} g(\omega_m) D(\omega_m) d\omega_m.$$

Dans la figure 4.3(c), nous utilisons cette propriété où g est la fonction de masquage G_1 que nous introduisons dans la section 4.2.3.

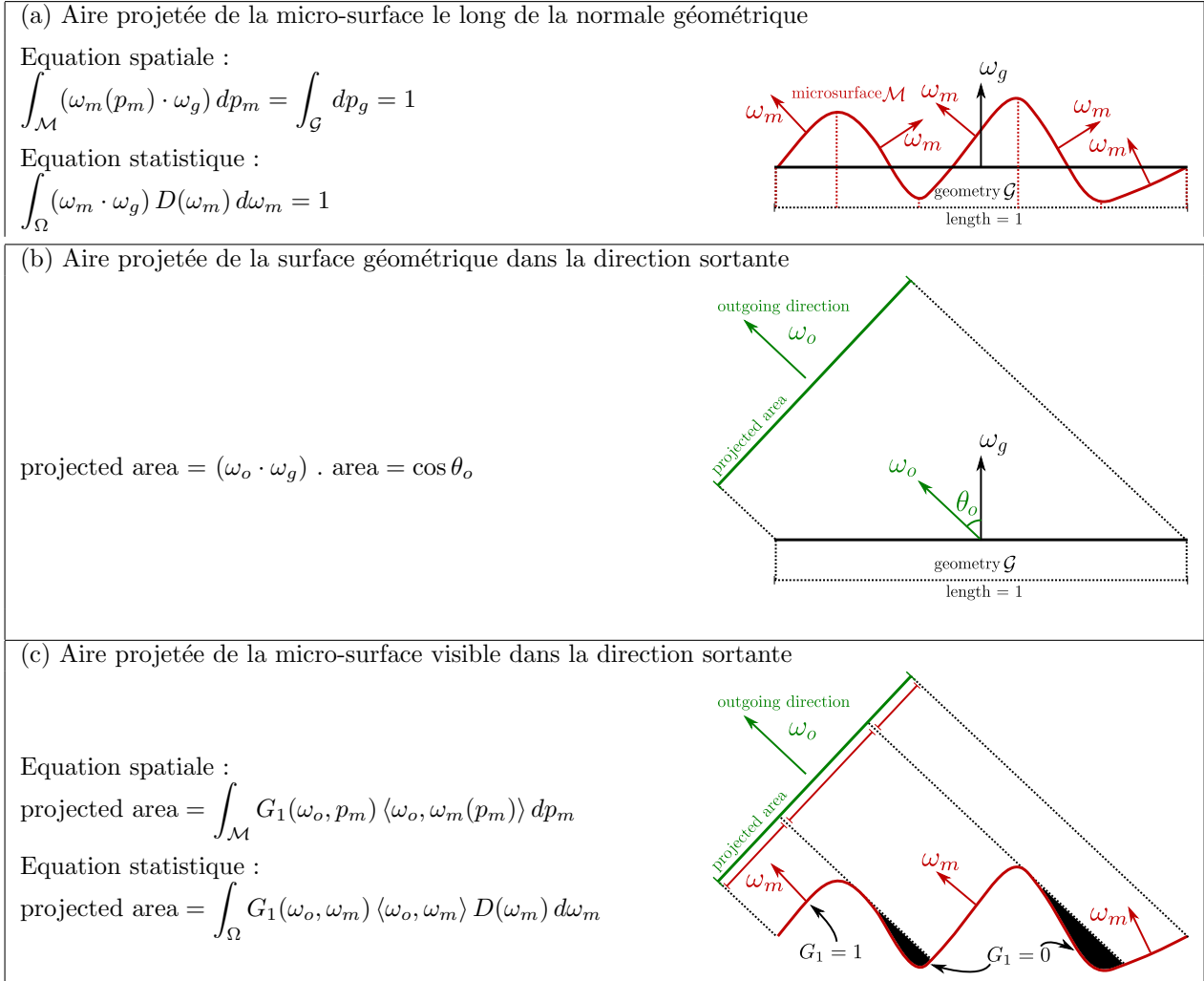


FIGURE 4.3 – Projections des micro-facettes.

4.2.3 Projections des micro-facettes

(a) Projection sur la surface géométrique. L'aire de la micro-surface projetée le long de la normale géométrique est l'aire de la surface géométrique (figure 4.3(a)), qui vaut 1m^2 , par convention. En conséquence, la projection de la distribution de normale sur la géométrie est normalisée :

$$\int_{\Omega} (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m = \int_{\mathcal{M}} (\omega_m(p_m) \cdot \omega_g) dp_m = \int_{\mathcal{G}} dp_g = 1\text{m}^2.$$

(b) Aire projetée de la surface géométrique. La surface géométrique est un élément de surface planaire dont l'aire projetée dans la direction sortante (figure 4.3(b)) est le cosinus de l'angle d'incidence θ_o :

$$\text{projected area} = (\omega_o \cdot \omega_g) \cdot \text{area} = \cos \theta_o \cdot 1\text{m}^2. \quad (4.2)$$

(c) Aire projetée de la micro-surface visible. L'aire projetée de la surface géométrique dans n'importe quelle direction est aussi l'aire projetée *visible* de la micro-surface dans cette direction (figure 4.3(c)). C'est la somme des aires projetées de l'ensemble des micro-facettes visibles. L'aire projetée d'une micro-facette de normale ω_m est le coefficient de projection $\langle \omega_o, \omega_m \rangle$. A noter que nous utilisons le produit scalaire clampé $\langle -, - \rangle$ car les normales auto-masquées ne sont pas visibles. De plus, les micro-facettes occluses par la micro-surface ne contribuent pas à l'aire projetée visible et n'interviennent pas dans cette somme. Cet effet est modélisé en multipliant par une *fonction de masquage spatiale* $G_1(\omega_o, p_m)$ à valeurs binaires : elle vaut 0 si le point p_m est masqué dans la direction ω_o et 1 s'il est visible :

$$\text{projected area} = \int_{\mathcal{M}} G_1(\omega_o, p_m) \langle \omega_o, \omega_m(p_m) \rangle dp_m.$$

La *fonction de masquage statistique* $G_1(\omega_o, \omega_m)$ à valeurs dans $[0, 1]$ donne la fraction de micro-facettes de normale ω_m qui sont visibles dans la direction ω_o :

$$G_1(\omega_o, \omega) = \frac{\int_{\mathcal{M}} \delta_{\omega}(\omega_m(p_m)) G_1(\omega_o, p_m) dp_m}{\int_{\mathcal{M}} \delta_{\omega}(\omega_m(p_m)) dp_m}.$$

L'équation statistique est donnée par :

$$\text{projected area} = \int_{\Omega} G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m. \quad (4.3)$$

4.2.4 Une contrainte sur la fonction de masquage

La figure 4.3 montre une propriété fondamentale de la théorie des micro-facettes : l'aire projetée visible de la micro-surface donnée par l'équation (4.3) est exactement l'aire projetée de la surface géométrique donnée par l'équation (4.2). Cette égalité impose une contrainte sur la fonction de masquage statistique, formalisée pour l'équation suivante :

$$\boxed{\cos \theta_o = \int_{\Omega} G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m.} \quad (4.4)$$

La fonction de masquage doit toujours satisfaire cette contrainte. Par contre, cette contrainte seule n'est pas suffisante pour déterminer G_1 entièrement, puisque pour une direction ω_o donnée, la fonction de masquage est une fonction 2D ($G_1(\omega_o, \omega_m)$ est définie pour chaque normale) et il y a un nombre infini de fonctions G_1 qui satisfont cette équation. De manière à réduire le nombre de solution à 1, on introduit une autre contrainte : on choisit un *profil de micro-surface*. En effet, la distribution de normales est comme un histogramme : elle décrit la proportion de chaque normale présente sur la micro-surface mais ne donne pas d'information sur la manière dont elles sont organisées. C'est cette information que donne le profil de surface qui, une fois choisi, détermine complètement la fonction de masquage dont la forme exacte peut être dérivée. Le profil de micro-surface impacte énormément la forme de la BRDF, comme le montre la figure 4.4. Dans la section 4.4, nous redécrivons les formes exactes de G_1 obtenues avec les profils de Smith et de surfaces à V-cavités.

4.2.5 Résumé

Une question récurrente concernant la fonction de masquage est : “*Parmi toutes les fonctions de masquage existantes, laquelle faut-il choisir ? Sont-elles toutes physiquement correctes ?*”

Dans cette section, nous avons montré que :

- L’aire projetée visible de la micro-surface vaut l’aire projetée de la surface géométrique, et ce, dans n’importe quelle direction.
- La fonction de masquage est contrainte par cette égalité. Une fonction de masquage physiquement correcte satisfait toujours l’équation (4.4).
- En revanche, la fonction de masquage n’est pas entièrement déterminée par l’équation (4.4).
- La fonction de masquage est entièrement déterminée une fois qu’un profil de micro-surface est choisi.
- Le profil de micro-surface impacte la forme de la BRDF.

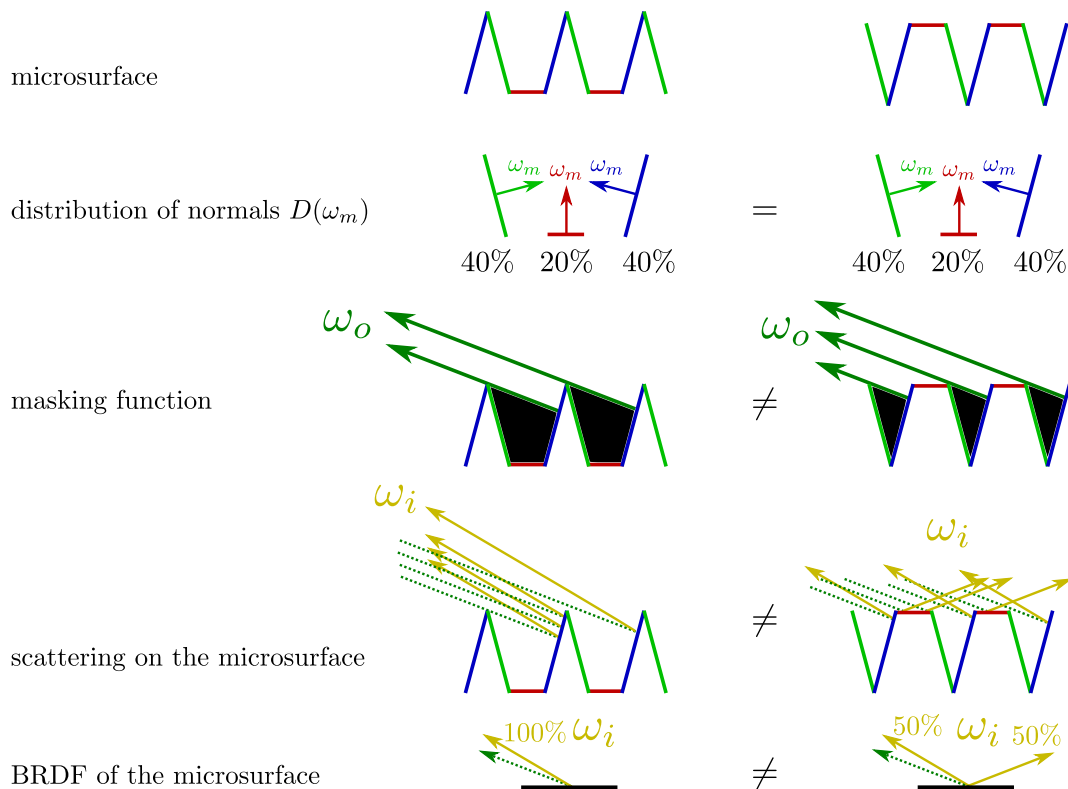


FIGURE 4.4 – Des micro-surfaces avec la même distribution de normales mais avec des profils différents produisent des BRDFs différentes.

4.3 Les modèles de BRDFs à micro-facettes

Dans cette section, nous définissons la distribution de normales visibles (4.3.1) et montrons comment les modèles à micro-facettes sont dérivables à partir de cette distribution (4.3.2) dans le cas général, et nous appliquons cette méthodologie aux cas particuliers des BRDFs à micro-facettes spéculaires (4.3.3) et à micro-facettes diffuses (4.3.4). Nous montrons que la fonction de masquage garantit la normalisation correcte de la distribution de normales visibles et nous discutons son lien avec la conservation de l'énergie dans la BRDF (4.3.5).

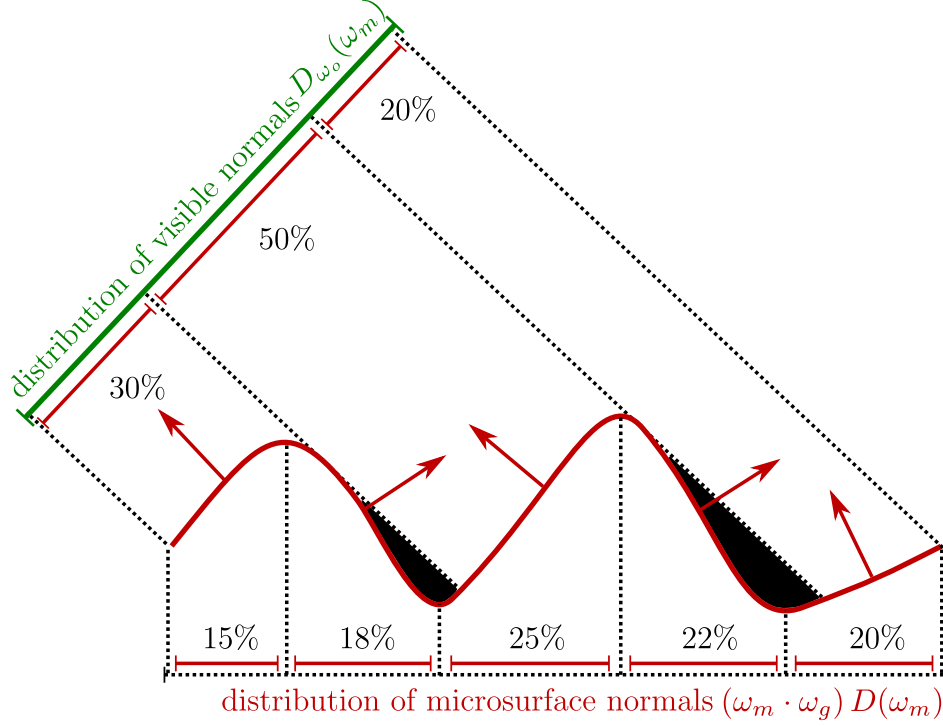


FIGURE 4.5 – La distribution des normales de la micro-surface $(\omega_m \cdot \omega_g) D(\omega_m)$ est une propriété intrinsèque à la surface, alors que la distribution des normales visibles $D_{\omega_o}(\omega_m)$ est une propriété qui dépend de la direction d'observation.

4.3.1 La distribution de normales visibles

Dans cette section, on montre que l'équation (4.1) peut se formuler dans le paradigme des micro-facettes par :

$$L(\omega_o, \mathcal{M}) = \frac{1}{\cos \theta_o} \int_{\Omega} L(\omega_o, \omega_m) G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m, \quad (4.5)$$

où $L(\omega_o, \omega_m)$ est la radiance sortante réfléchiée par les micro-facettes de normale ω_m , et le facteur $\frac{1}{\cos \theta_o}$ a pour rôle de renormaliser l'intégrale par l'aire projetée de la surface géométrique. On peut voir que la radiance sortante de la surface est la somme des radiances sortantes de chaque micro-facette pondérées par ce qu'on appelle la *distribution de normales visibles*, illustrées en Figure 4.5. C'est la distribution des normales de la micro-surface pondérées par leurs aires projetées (le cosinus clampé) et par la fonction de masquage :

$$D_{\omega_o}(\omega_m) = \frac{G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m)}{\cos \theta_o}, \quad (4.6)$$

Il est important que la distribution de normales visibles $D_{\omega_o}(\omega_m)$ soit normalisée, puisque nous l'utilisons comme une fonction de pondération pour moyenner des radiances :

$$L(\omega_o, \mathcal{M}) = \int_{\Omega} L(\omega_o, \omega_m) D_{\omega_o}(\omega_m) d\omega_m, \quad (4.7)$$

et, comme expliqué en section 4.2.1 et en figure 4.1, moyenner des radiances n'est valide qu'à condition que la fonction de pondération soit normalisée. Cette dernière équation est bien définie puisque l'intégrale au dénominateur de l'équation (4.1), qui assure une normalisation correcte, est maintenant représentée par la fonction de masquage G_1 . En effet, en utilisant le résultat de l'équation (4.4), nous pouvons substituer $\cos \theta_o$ dans l'équation (4.6) et vérifier que la distribution de normales visibles est normalisée :

$$\begin{aligned} \int_{\Omega} D_{\omega_o}(\omega_m) d\omega_m &= \int_{\Omega} \frac{G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m)}{\cos \theta_o} d\omega_m \\ &= \frac{\int_{\Omega} G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m}{\int_{\Omega} G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m} \\ &= 1, \end{aligned} \quad (4.8)$$

et la radiance sortante moyennée des équations (4.5) et (4.7) peut être exprimée sous la même forme que dans l'équation (4.1), faisant ressortir la normalisation correcte :

$$L(\omega_o, \mathcal{M}) = \frac{\int_{\Omega} L(\omega_o, \omega_m) G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m}{\int_{\Omega} G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m}.$$

4.3.2 Construction d'une BRDF à micro-facettes

Dans cette section, nous construisons la BRDF en partant de la distribution de normales visibles. La radiance $L(\omega_o, \omega_m)$ associée à chaque micro-facette peut être exprimée en fonction de la *micro-BRDF* $\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m)$, en intégrant sur l'ensemble des directions incidentes Ω_i (nous gardons la notation Ω pour l'espace des normales) :

$$L(\omega_o, \omega_m) = \int_{\Omega_i} dL(\omega_o, \omega_m) = \int_{\Omega_i} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_i, \omega_m \rangle L(\omega_i) d\omega_i,$$

où la micro-BRDF est définie comme étant le ratio différentiel de la radiance sortant et de l'irradiance incidente :

$$dL(\omega_o, \omega_m) = \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_i, \omega_m \rangle L(\omega_i) d\omega_i.$$

Nous utilisons cette définition de la micro-BRDF pour dériver l'équation (4.7) :

$$\begin{aligned} dL(\omega_o, \mathcal{M}) &= \int_{\Omega} dL(\omega_o, \omega_m) D_{\omega_o}(\omega_m) d\omega_m \\ &= L(\omega_i) d\omega_i \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_i, \omega_m \rangle D_{\omega_o}(\omega_m) d\omega_m, \end{aligned} \quad (4.9)$$

où $L(\omega_i) d\omega_i$ peut être sorti de l'intégrale puisqu'il ne dépend pas de ω_m . Puisque la macro-BRDF est définie par l'équation :

$$dL(\omega_o, \mathcal{M}) = \rho(\omega_o, \omega_i) \cos \theta_i L(\omega_i) d\omega_i,$$

nous arrivons à la forme suivante :

$$\begin{aligned} \rho(\omega_o, \omega_i) &= \frac{dL(\omega_o, \mathcal{M})}{\cos \theta_i L(\omega_i) d\omega_i} \\ &= \frac{1}{\cos \theta_i} \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_i, \omega_m \rangle D_{\omega_o}(\omega_m) d\omega_m. \end{aligned}$$

En substituant $D_{\omega_o}(\omega_m)$ de l'équation (4.6) nous obtenons :

$$\begin{aligned}\rho(\omega_o, \omega_i) &= \frac{1}{\cos \theta_o \cos \theta_i} \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o, \omega_m) D(\omega_m) d\omega_m \\ &= \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o, \omega_m) D(\omega_m) d\omega_m.\end{aligned}$$

Une observation importante est que cette équation modélise uniquement comment les rayons sont réfléchis juste après la première réflexion spéculaire et *avant* de quitter la micro-surface (figure 4.6(b)). Mais un modèle de BRDF décrit comment les rayons sont distribués *après* avoir quitté la micro-surface. La distribution avant et après avoir quitté la micro-surface n'est pas la même car certains rayons peuvent encore intersecter la micro-surface et être réfléchis dans une autre direction (figure 4.6(d)). Puisque le modèle de BRDF dérivé ici ne modélise que le premier rebond sur la micro-surface, les rayons subissant plusieurs rebonds ne sont pas pris en compte (figure 4.6(c)) et ne sont pas incorporés dans ce modèle. Cela est fait en introduisant une fonction d'*ombrage*. La fonction de masquage G_1 est remplacée par une fonction de masquage et d'ombrage G_2 :

$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m. \quad (4.10)$$

Nous instancions cette équation pour le cas spécial où les micro-facettes sont des miroirs parfaits (4.3.3) ou des diffuseurs parfaits (4.3.4).

4.3.3 Construction d'une BRDF à micro-facettes spéculaires

Lorsque les micro-facettes sont des miroirs parfaits, la micro-BRDF dans ce cas est donnée par :

$$\begin{aligned}\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) &= \left\| \frac{\partial \omega_h}{\partial \omega_i} \right\| \frac{F(\omega_o, \omega_h) \delta_{\omega_h}(\omega_m)}{|\omega_i \cdot \omega_h|} \\ &= \frac{F(\omega_o, \omega_h) \delta_{\omega_h}(\omega_m)}{4 |\omega_i \cdot \omega_h|^2},\end{aligned} \quad (4.11)$$

où $\left\| \frac{\partial \omega_h}{\partial \omega_i} \right\| = \frac{1}{4 |\omega_i \cdot \omega_h|}$ est la jacobien de l'opérateur de réflexion spéculaire [WMLT07], et F est le terme de Fresnel. Dans l'équation (4.10), en substituant $\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m)$ de l'équation (4.11) nous obtenons :

$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\Omega} \frac{F(\omega_o, \omega_h) \delta_{\omega_h}(\omega_m)}{4 |\omega_i \cdot \omega_h|^2} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m.$$

La distribution de dirac $\delta_{\omega_h}(\omega_m)$ nous permet de remplacer l'intégrale par l'intégrande évaluée en $\omega_m = \omega_h$, et le fait que $\omega_o \cdot \omega_h = \omega_i \cdot \omega_h$ permet de réduire l'expression à :

$$\rho(\omega_o, \omega_i) = \frac{F(\omega_o, \omega_h) G_2(\omega_o, \omega_i, \omega_h) D(\omega_h)}{4 |\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|}. \quad (4.12)$$

Nous sommes arrivés à l'équation classique du modèle de BRDF à micro-facettes spéculaires [WMLT07].

4.3.4 Construction d'une BRDF à micro-facettes diffuses

Lorsque les micro-facettes sont des diffuseurs parfaits, la micro-BRDF est constante :

$$\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) = \frac{1}{\pi}. \quad (4.13)$$

Dans l'équation (4.10), en substituant $\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m)$ de l'équation (4.13) nous obtenons :

$$\rho(\omega_o, \omega_i) = \frac{1}{\pi} \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\Omega} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m. \quad (4.14)$$

Cette équation n'a pas de forme analytique mais peut être pré-calculée ou approximée. C'est ce que font Oren et Nayar [ON94] en prenant pour D est une gaussienne sphérique (à ne pas confondre avec une distribution de Beckmann) et pour G_2 la fonction de masquage et d'ombrage du modèle à V-cavités.

4.3.5 Le test de normalisation de la BRDF

Le White Furnace Test. La bidirectional scattering distribution function (BSDF) s est la somme de la bidirectional reflectance distribution function (BRDF) ρ , définie sur l'hémisphère supérieure, et de la bidirectional transmittance distribution function (BTDF) t définie sur l'hémisphère inférieure :

$$s(\omega_o, \omega_i) = \rho(\omega_o, \omega_i) + t(\omega_o, \omega_i).$$

Sur une surface parfaitement blanche ne dissipant jamais l'énergie en chaleur, l'énergie des rayons serait parfaitement préservée. Une propriété importante devant être vérifiée par les modèles à micro-facettes est que, lorsque l'absorption de la surface vaut 0, la distribution des rayons réfléchis et transmis est normalisée :

$$\int_{\Omega_i} s(\omega_o, \omega_i) |\omega_g \cdot \omega_i| d\omega_i = 1.$$

Si le terme de Fresnel vaut toujours 1, les rayons ne sont jamais transmis (ils ne pénètrent jamais la surface), la BTDF vaut toujours $t = 0$, et le modèle de BSDF est alors entièrement déterminé par la BRDF (i.e., $s = \rho$). Dans ce cas, les rayons sont tous réfléchis sans perte d'énergie et leur distribution est normalisée. Cela est modélisé par l'équation du *White Furnace Test* :

$$\int_{\Omega_i} \rho(\omega_o, \omega_i) |\omega_g \cdot \omega_i| d\omega_i = 1.$$

Intuitivement, cette équation représente le fait que les rayons partant de la direction sortante (figure 4.6(a)) seraient réfléchis une ou plusieurs fois avant de finalement quitter la micro-surface (figure 4.6(d)). En revanche, le modèle de BRDF dérivé ci-dessus ne modélise pas les rebonds multiples ; les rayons rebondissant plusieurs fois sont supprimés de la BRDF par la fonction d'ombrage, comme illustré en figure 4.6(c) et écrit en section 4.3.2. C'est la raison pour laquelle ce modèle de BRDF ne s'intègre pas à 1 et ne satisfait pas l'équation du *White Furnace Test*.

Le Weak White Furnace Test. Le *White Furnace Test* ne peut pas être utilisé pour valider le modèle de BRDF qui n'incorpore que le premier rebond. Mais nous pouvons imaginer un autre test, moins restrictif, qui doit être satisfait par ce modèle. Voici ce que nous proposons : nous pouvons vérifier que la distribution de rayons réfléchis juste après le premier rebond et *avant* de quitter la micro-surface est normalisée (figure 4.6(b)). Ceci peut être réalisé en remplaçant la fonction de masquage et d'ombrage par la fonction de masquage uniquement : $G_2(\omega_o, \omega_i, \omega_h) = G_1(\omega_o, \omega_h)$. Sans le terme de Fresnel et la fonction d'ombrage, la BRDF de l'équation (4.12) devient :

$$\rho(\omega_o, \omega_i) = \frac{G_1(\omega_o, \omega_h) D(\omega_h)}{4 |\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|},$$

et après simplification des $|\omega_g \cdot \omega_i|$, notre *Weak White Furnace Test* est donné par l'équation :

$$\boxed{\int_{\Omega_i} \frac{G_1(\omega_o, \omega_h) D(\omega_h)}{4 |\omega_g \cdot \omega_o|} d\omega_i = 1.} \quad (4.15)$$

Nous pouvons définir le même test pour les BRDFs à micro-facettes diffuses, en remplaçant $G_2(\omega_o, \omega_i, \omega_h) = G_1(\omega_o, \omega_h)$ dans l'équation (4.14) et en intégrant sur l'ensemble des directions incidentes :

$$\begin{aligned} & \int_{\Omega_i} \frac{1}{\pi} \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\Omega} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o, \omega_m) D(\omega_m) d\omega_m |\omega_g \cdot \omega_i| d\omega_i \\ &= \frac{1}{\pi} \frac{1}{|\omega_g \cdot \omega_o|} \int_{\Omega_i} \int_{\Omega} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o, \omega_m) D(\omega_m) d\omega_m d\omega_i = 1. \end{aligned} \quad (4.16)$$

Nous verrons dans la section 4.4 quels sont les modèles qui satisfont ce test.

4.3.6 Résumé

Une question récurrente concernant la normalisation des BRDFs est la suivante : “*Les BRDFs à micro-facettes ne s'intègrent pas à 1, ne devraient-elles pas être parfaitement normalisées ?*”

Dans cette section, nous avons répondu à cette question en développant les idées suivantes :

- Les BRDFs à micro-facettes sont construites à partir de la distribution de normales visibles.
- La distribution de normales visibles doit être normalisée pour assurer que la BRDF conserve l'énergie.
- Le coefficient de normalisation de la distribution de normales visibles est la fonction de masquage.
- Les BRDFs à micro-facettes devraient théoriquement s'intégrer à 1 lorsque la surface n'est pas absorbante.
- La fonction d'ombrage dans les BRDFs à micro-facettes est utilisée pour séparer l'énergie réfléchie par le premier rebond de l'énergie réfléchie après des rebonds multiples. L'ombrage met à 0 l'énergie réfléchie après plusieurs rebonds et produit une BRDF artificiellement non normalisée en l'absence d'un terme pour modéliser les rebonds multiples.
- La forme standard des BRDFs à micro-facettes est normalisée sans le terme de Fresnel et sans l'ombrage. Les fonctions de masquage physiquement correctes satisfont toujours les équations (4.15) et (4.16). C'est ce qu'on appelle le *Weak White Furnace Test*.

A noter que le Weak White Furnace Test que nous proposons, dans lequel l'ombrage n'est pas incorporé, est un moyen simple de vérifier que la fonction de masquage est bien définie. C'est une contrainte forte sur la fonction de masquage dont on peut se servir en pratique pour déboguer ses équations ou son code. Mais il est important de comprendre que cela ne signifie pas que les modèles de BRDF devraient être utilisés sans ombrage. L'ombrage est un élément essentiel de la BRDF qui ne modélise que le premier rebond et ne peut pas être supprimé de ce modèle.

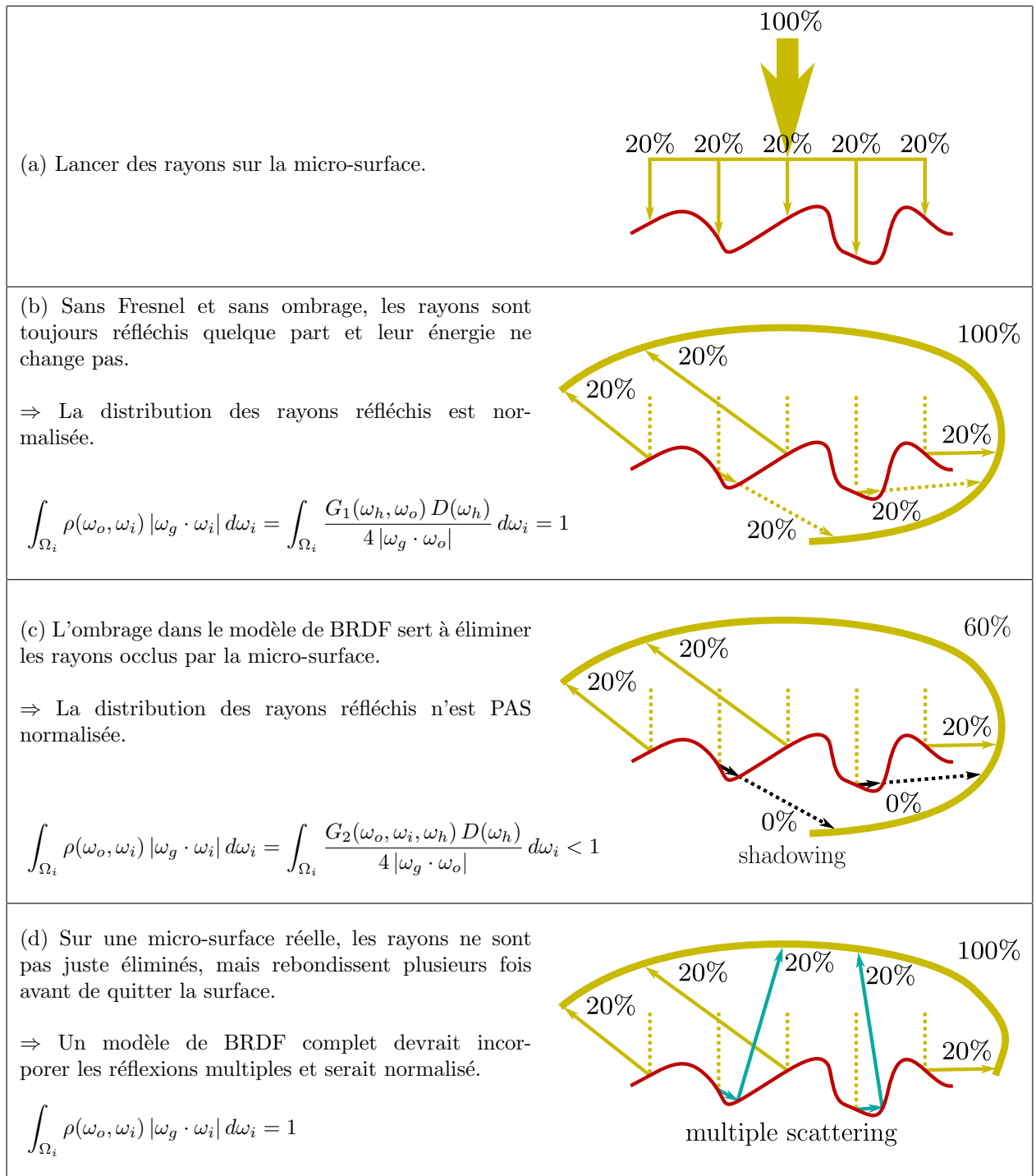


FIGURE 4.6 – La normalisation des BRDFs à micro-facettes (équations données pour la BRDF à micro-facettes spéculaires).

4.4 Choisir un profil de micro-surface

Dans les sections 4.2 and 4.3 nous avons dérivé des résultats généraux sur la fonction de masquage, donnés par les équations (4.4), (4.8) et (4.15), sans faire d'hypothèses sur le type de micro-surface considéré. Dans cette section, nous dérivons les fonctions de masquage exactes pour les profils de micro-surface des modèles de Smith (4.4.1) et à base de V-cavités (4.4.2), et nous discutons leurs propriétés.

4.4.1 Le profil de micro-surface de Smith

Indépendance des normales et du masquage. Le profil de micro-surface de Smith suppose que la micro-surface n'est pas auto-corrélée, c'est-à-dire que la micro-surface ressemble plus à une soupe volumique 3D de micro-facettes qu'à une surface continue, comme illustré en figure 4.7, et où les hauteurs et les normales des points de la micro-surfaces sont modélisés par des variables aléatoires indépendantes.

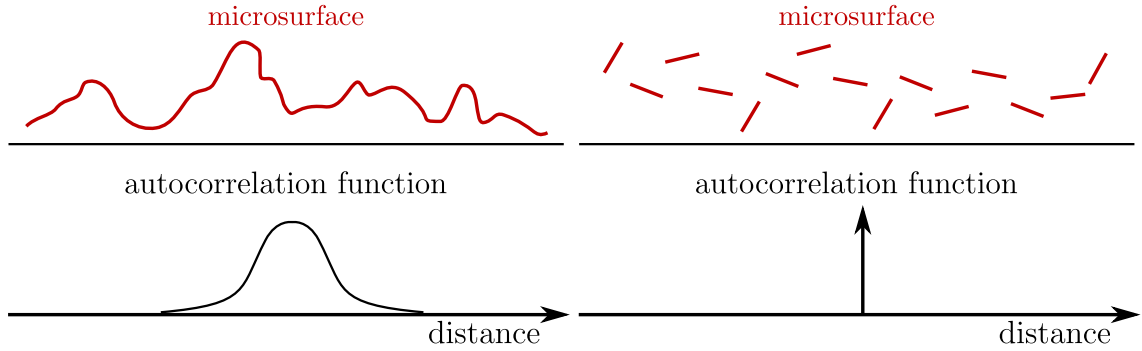


FIGURE 4.7 – Des micro-surfaces et leurs fonctions d'auto-corrélation. (gauche) Une micro-surface continue avec une large fonction d'auto-corrélation. (droite) Une micro-surface où les micro-facettes ne sont pas corrélées à leurs voisines, comme modélisé par Smith.

La conséquence de ce modèle est que la probabilité de masquage $G_1(\omega_o, \omega_m)$ d'une micro-facette est indépendante de la normale de cette micro-facette ω_m , lorsque celle-ci n'est pas auto-masquée ($\langle \omega_o, \omega_m \rangle > 0$). L'intuition derrière cette propriété est que la normale ω_m est une propriété *locale* de la micro-facette, alors que l'occlusion potentielle par un autre point de la micro-surface est une propriété *distante* de la micro-facette. Puisque la micro-surface n'est pas auto-corrélée, les propriétés locales et distantes sont indépendantes et la fonction de masquage peut-être exprimée sous la forme séparable :

$$G_1(\omega_o, \omega_m) = G_1^{\text{local}}(\omega_o, \omega_m) G_1^{\text{dist}}(\omega_o), \quad (4.17)$$

où la fonction de masquage locale est l'auto-masquage dû à l'orientation de la micro-facette :

$$G_1^{\text{local}}(\omega_o, \omega_m) = \chi^+(\omega_o \cdot \omega_m),$$

et la fonction de masquage distante $G_1^{\text{dist}}(\omega_o)$ est la probabilité d'occlusion par un autre point de la micro-surface et est indépendante de l'orientation locale ω_m de la micro-facette.

Dérivation de la fonction de masquage. En développant la fonction de masquage de l'équation (4.17) dans l'équation (4.4), nous obtenons :

$$\begin{aligned}
\cos \theta_o &= \int_{\Omega} G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m \\
&= \int_{\Omega} G_1^{\text{local}}(\omega_o, \omega_m) G_1^{\text{dist}}(\omega_o) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m \\
&= \int_{\Omega} \chi^+(\omega_o \cdot \omega_m) G_1^{\text{dist}}(\omega_o) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m \\
&= G_1^{\text{dist}}(\omega_o) \int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m,
\end{aligned}$$

où nous avons sorti $G_1^{\text{dist}}(\omega_o)$ de l'intégrale puisqu'il ne dépend pas de ω_m et nous avons supprimé $\chi^+(\omega_o \cdot \omega_m)$ qui est redondant avec le produit scalaire clampé qui vaut déjà 0 lorsque $\omega_o \cdot \omega_m < 0$. La fonction de masquage pour les normales non auto-masquées depuis la direction sortante ω_o est donnée par :

$$G_1^{\text{dist}}(\omega_o) = \frac{\cos \theta_o}{\int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m}, \quad (4.18)$$

et la fonction de masquage vaut :

$$G_1(\omega_o, \omega_m) = \chi^+(\omega_o \cdot \omega_m) \frac{\cos \theta_o}{\int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m}. \quad (4.19)$$

Ceci est la forme intégrale de la fonction de masquage exacte sous l'hypothèse d'indépendance des normales et du masquage, présentée par Ashikhmin et al. [APS00]. Ils utilisent cette expression intégrale pour pré-calculer la fonction d'ombrage et la stocker dans une table.

La fonction de masquage de Smith. En changeant l'espace d'intégration défini sur la sphère à l'espace des pentes (la dérivation complète est disponible dans l'annexe A.1), l'équation (4.18) devient :

$$G_1^{\text{dist}}(\omega_o, \omega_m) = \frac{1}{1 + \Lambda(\omega_o)},$$

et donc l'équation (4.19) peut s'écrire :

$$G_1(\omega_o, \omega_m) = \frac{\chi^+(\omega_o \cdot \omega_m)}{1 + \Lambda(\omega_o)}, \quad (4.20)$$

où $\frac{1}{1 + \Lambda(\omega_o)}$ est la forme généralisée de la fonction d'ombrage de Smith [Bro80, WMLT07], pour laquelle des formes closes existent pour plusieurs distributions de normales paramétriques (nous rappelons ses formes explicites pour les distribution de Beckmann et GGX en section 4.5). Cette dérivation montre que, sous l'hypothèse d'indépendance des normales et du masquage, la fonction de masquage de Smith est exacte.

Propriétés. Pourtant, si nous comparions la fonction analytique de Smith avec des données mesurées, nous trouverions que les prédictions du modèle sont précises mais néanmoins pas exactes. En effet, Smith compare sa formule à des données mesurées sur des échantillons de surfaces réelles et trouve que sa fonction était une bonne approximation, mais juste une approximation. Mais l'approximation ne vient pas de sa dérivation, car, sous les hypothèses de son modèle de micro-surface, elle est exacte. L'approximation réside justement dans la modélisation de surfaces réelles avec des modèles statistiques idéaux : surfaces gaussiennes et l'hypothèse d'indépendance.

Le modèle de micro-surface non auto-corrélée de Smith rappelle les modèles de “metal flakes”, qu'on trouve dans certaines peintures métalliques [RMS+08]. Mais des surfaces continues réelles sont auto-corrélées.

Bourlier et al. comparent la fonction de masquage de Smith avec des mesures réalisées sur des échantillons numériques avec différentes fonctions d’auto-corrélation (gaussiennes et lorentziennes) [BSB00, BBS02]. La conclusion de leur recherche est que l’erreur introduite en négligeant la corrélation dans un modèle de surface stochastique est en moyenne insignifiante et observable seulement à des angles d’incidence tels que $\tan(\theta)/\alpha > 0.5$, où $\sigma^2 = \frac{\alpha^2}{2}$ est la variance de la distribution des pentes. La fonction de masquage de Smith tend à surestimer le masquage dans ce cas. Sachant que la fonction de Smith est généralement précise, même avec des surfaces auto-corrélées, et sachant qu’il n’existe pas de forme analytique de la fonction de masquage sur des surfaces corrélées, il semble raisonnable de s’en tenir à la fonction de masquage de Smith pour des applications d’image de synthèse. En revanche, comme souligné par Ashikhmin et al. [APS00], l’effet de l’auto-corrélation sur des surfaces non stochastiques, avec des motifs répétitifs ou structurés (par exemple du tissu), peut s’avérer important et doit être incorporé dans des modèles dédiés.

Les fonctions de masquage de Smith moyennées. Smith dérive sa fonction de masquage moyennée sur différentes quantités de la micro-surface, comme les hauteurs et les normales [Smi67]. La fonction de masquage $G_1(\omega_o, \omega_m)$ présentée en équation (4.20) est la forme moyennée sur les hauteurs et elle est celle qui doit être utilisée dans le modèle de BRDFs à micro-facettes basé sur la géométrie optique que nous présentons dans ce chapitre. En effet, puisque les hauteurs sont indépendantes des normales, seules variables qui comptent dans ce modèle de BRDF, on peut moyenner leurs contributions. En revanche, les normales n’interviennent pas toutes de la même manière dans le modèle : les normales auto-masquées ne sont pas prises en compte. Dans ce modèle de BRDF, seulement ce qui est visible compte, car la BRDF modélise la radiance visible et mesurable. Si quelque chose existe sur la micro-surface mais n’est pas visible, elle ne doit pas être considérée dans cette BRDF. Dans ce modèle de BRDF, on se pose la question : “*Quelle est la proportion de normales non auto-masquées qui sont masquées ?*”

Dans ce cas, pourquoi Smith a-t-il dérivé une forme de sa fonction de masquage moyennée sur les normales, si elle n’est pas utile pour les BRDFs ? Il voulait en fait répondre à la question : “*Quelle est la proportion de normales qui sont masquées sur l’ensemble de la surface ?*” La réponse à cette question est importante pour d’autres modèles de BRDFs, par exemple celui de He et al. [HTSG91] basé sur l’optique ondulatoire où l’ensemble de la surface contribue à la réflexion dans chaque direction. Dans le modèle de BRDF entièrement basé sur la géométrie optique que nous discutons dans ce chapitre et que nous utiliserons dans le reste de cette thèse, les normales auto-masquées ne contribuent pas à la BRDF (par construction géométrique). Il n’est donc pas nécessaire de les introduire dans le calcul de la fonction de masquage.

4.4.2 Le profil de micro-surface à V-cavités

Dans cette section, nous discutons le profil de micro-surface à V-cavités [CT82, ON94], qui est l'alternative la plus commune à celui de Smith. La figure 4.8 illustre le modèle de BRDF avec une micro-surface à V-cavités. Plutôt que de modéliser le comportement optique d'une unique micro-surface avec une distribution de normales, ce modèle suppose des micro-surfaces séparées et moyenne leurs comportements optiques. Chaque micro-surface est composée de deux normales $\omega_m = (x_m, y_m, z_m)$ et $\omega_m' = (-x_m, -y_m, z_m)$ et la contribution de chaque micro-surface est pondérée par $\langle \omega_m, \omega_g \rangle D(\omega_m)$ dans la BRDF.

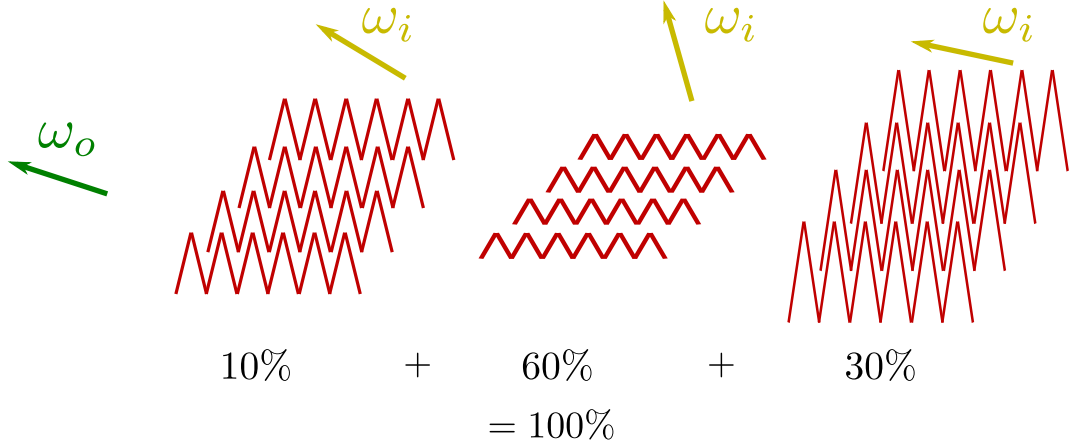


FIGURE 4.8 – Le modèle à V-cavités. Plutôt que de modéliser le comportement optique d'une unique micro-surface avec une distribution de normales, ce modèle suppose des micro-surfaces séparées et moyenne leurs comportements optiques.

Une démonstration géométrique est habituellement présentée pour dériver la fonction de masquage de la micro-surface à V-cavité. Cette démonstration est laborieuse car elle nécessite de longs calculs trigonométriques. Nous pouvons dériver le même résultat en appliquant la contrainte de conservation de l'aire de la micro-surface visible projetée, comme en section 4.2.3. Les micro-surfaces à V-cavités ont seulement deux normales symétriques ω_m et ω_m' . La distribution de normales de cette micro-surface est :

$$D(\omega) = \frac{1}{2} \frac{\delta_{\omega_m}(\omega)}{\omega_m \cdot \omega_g} + \frac{1}{2} \frac{\delta_{\omega_m'}(\omega)}{\omega_m' \cdot \omega_g}.$$

Nous vérifions que la normalisation est correcte :

$$\begin{aligned} \int_{\Omega} \langle \omega, \omega_g \rangle D(\omega) d\omega &= \int_{\Omega} \langle \omega, \omega_g \rangle \left(\frac{1}{2} \frac{\delta_{\omega_m}(\omega)}{\omega_m \cdot \omega_g} + \frac{1}{2} \frac{\delta_{\omega_m'}(\omega)}{\omega_m' \cdot \omega_g} \right) d\omega \\ &= \frac{1}{2} \frac{\omega_m' \cdot \omega_g}{\omega_m' \cdot \omega_g} + \frac{1}{2} \frac{\omega_m' \cdot \omega_g}{\omega_m' \cdot \omega_g} \\ &= \frac{1}{2} + \frac{1}{2} = 1. \end{aligned}$$

Pour dériver la fonction de masquage, nous utilisons l'équation (4.4) :

$$\begin{aligned} \cos \theta_o &= \int_{\Omega} G_1(\omega_o, \omega) \langle \omega_o, \omega \rangle D(\omega) d\omega \\ &= \frac{1}{2} G_1(\omega_o, \omega_m) \frac{\langle \omega_o, \omega_m \rangle}{\omega_m \cdot \omega_g} + \frac{1}{2} G_1(\omega_o, \omega_m') \frac{\langle \omega_o, \omega_m' \rangle}{\omega_m' \cdot \omega_g}. \end{aligned}$$

Il y a deux configurations possibles, comme le montre la figure 4.9. Dans le premier cas, les deux normales sont visibles et il n'y a pas de masquage : $G_1(\omega_o, \omega_m) = 1$ et $G_1(\omega_o, \omega_m') = 1$. Dans le second cas, ω_m' est

auto-masquée ($G_1(\omega_o, \omega_m') = 0$) et nous avons :

$$\cos \theta_o = \frac{1}{2} G_1(\omega_o, \omega_m) \frac{\langle \omega_o, \omega_m \rangle}{\omega_m \cdot \omega_g},$$

dont la solution est :

$$\begin{aligned} G_1(\omega_o, \omega_m) &= 2 \frac{\cos \theta_o (\omega_m \cdot \omega_g)}{\langle \omega_o, \omega_m \rangle} \\ &= 2 \frac{(\omega_m \cdot \omega_g)(\omega_o \cdot \omega_g)}{\langle \omega_o, \omega_m \rangle}. \end{aligned}$$

Le résultat de ces deux configurations peut être exprimé dans une formule unique :

$$G_1(\omega_o, \omega_m) = \min \left(1, 2 \frac{(\omega_m \cdot \omega_g)(\omega_o \cdot \omega_g)}{\langle \omega_o, \omega_m \rangle} \right), \quad (4.21)$$

qui est la fonction de masquage de Cook et Torrance [CT82].

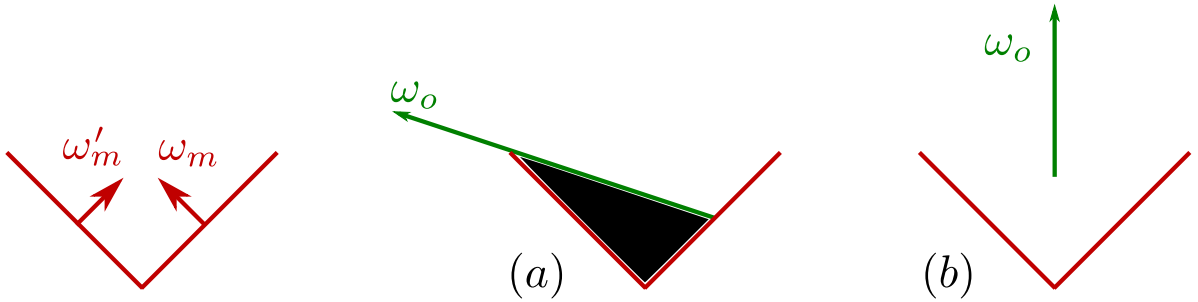


FIGURE 4.9 – Configurations de masquage sur une micro-surface à V-cavités. Soit une des normales est auto-masquée et l'autre est partiellement masquée (a), soit les deux normales sont visibles et la fonction de masquage vaut 1.

Validation. Nous vérifions que ce modèle satisfait l'équation (4.8), c'est-à-dire que la distribution des normales visibles est normalisée. Nous substituons G_1 de l'équation 4.21 :

$$D_{\omega_o}(\omega_m) = \min \left(1, 2 \frac{(\omega_m \cdot \omega_g)(\omega_o \cdot \omega_g)}{\langle \omega_o, \omega_m \rangle} \right) \frac{\langle \omega_o, \omega_m \rangle D(\omega_m)}{\cos \theta_o}.$$

Cette forme est compliquée à étudier directement à cause de l'opérateur $\min(1, -)$. Nous allons voir que la différence principale entre les modèles de Smith et à V-cavités se produit pour des incidences rasantes. Pour simplifier l'étude et aller directement à l'essentiel, nous allons concentrer la dérivation sur ce qui se passe aux angles rasants, lorsque $\theta_o \approx \frac{\pi}{2}$. En effet, aux angles rasants, nous sommes toujours dans la configuration (a) de la figure 4.9, où une des micro-facettes est masquée. Dans ce cas, on peut éliminer l'opérateur $\min(1, -)$:

$$\begin{aligned} D_{\omega_o}(\omega_m) &= 2 \frac{(\omega_m \cdot \omega_g)(\omega_o \cdot \omega_g)}{\langle \omega_o, \omega_m \rangle} \frac{\langle \omega_o, \omega_m \rangle D(\omega_m)}{\cos \theta_o} \\ &= 2 \chi^+(\omega_o \cdot \omega_m) (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m. \end{aligned} \quad (4.22)$$

A noter que le produit scalaire clampé $\langle \omega_o, \omega_m \rangle$ se simplifie mais la fonction de Heaviside $\chi^+(\omega_o \cdot \omega_m)$ reste pour assurer que les normales auto-masquées soient toujours supprimées de la distribution. Nous validons ce résultat en vérifiant que la distribution de normales visibles est normalisée. Nous calculons :

$$\int_{\Omega} D_{\omega_o}(\omega_m) d\omega_m = 2 \int_{\Omega} \chi^+(\omega_o \cdot \omega_m) (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m.$$

Puisque la direction sortante est presque orthogonale à ω_g à incidence rasante, elle tronque l'intégrale presque au milieu de la distribution. De plus, une surface à V-cavités suppose que la distribution de normales soit symétrique : $D(\omega_m) = D(\omega_m')$. Cela implique que la fonction de Heaviside coupe la distribution de normales en deux parties égales et nous obtenons :

$$\begin{aligned} \int_{\Omega} \chi^+(\omega_o \cdot \omega_m) (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m &= \frac{1}{2} \int_{\Omega} (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m \\ &= \frac{1}{2}, \end{aligned}$$

où nous utilisons le fait que la distribution de normales est normalisée : $\int_{\Omega} (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m = 1$. Nous obtenons :

$$\begin{aligned} \int_{\Omega} D_{\omega_o}(\omega_m) d\omega_m &= 2 \int_{\Omega} \chi^+(\omega_o \cdot \omega_m) (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m \\ &= 2 \frac{1}{2} = 1. \end{aligned}$$

Cela montre que la distribution de normales visibles de l'équation 4.22 est normalisée pour des angles d'incidence rasants. Une dérivation plus technique peut montrer que la distribution est normalisée pour n'importe quel angle d'incidence. Une autre manière de valider ce modèle est d'utiliser le Weak White Furnace Test : nous évaluons l'équation 4.15 en utilisant G_1 de l'équation 4.21. Le résultat de l'intégrale est toujours 1 et le modèle d'ombrage avec des micro-surfaces à V-cavités conserve l'énergie.

Propriétés. Néanmoins, alors que la distribution de normales visibles des V-cavités est mathématiquement bien définie, ce profil de micro-surface n'est pas physiquement plausible et modélise une surface qui est physiquement non réaliste.

Il y a deux types de normales aux angles rasants : celles qui sont auto-masquées et supprimées par la fonction de Heaviside et les autres, qui sont visibles, sont pondérées par $(\omega_m \cdot \omega_g) D(\omega_m)$. A noter que le facteur $(\omega_m \cdot \omega_g)$ est le jacobien de la projection d'une micro-facette sur la surface géométrique, comme illustré dans la figure 4.3(a). De fait, les micro-facettes sont pondérées exactement comme si elles étaient projetées sur la surface géométrique avant d'être projetées dans la direction sortante ω_o . En conséquence, nous modélisons des micro-facettes "plates" aux angles rasants : elles peuvent perturber la direction de réflexion de la lumière mais elles n'existent pas géométriquement. La raison pour laquelle ce profil de micro-surface n'est pas réaliste est qu'il se comporte plus comme une normal map qu'une height map aux angles rasants, comme illustré dans la figure 4.10.

Cet effet était attendu : plutôt que de modéliser une seule micro-surface, le modèle à V-cavités modélise une micro-surface par paire de normales et moyenne leurs contributions. Mais sur une unique micro-surface, les normales les plus visibles occupent plus d'aire projetée que les normales moins visibles et leur pondération est plus importante. Cette concurrence pour la visibilité entre les micro-facettes n'existe pas dans le modèle à V-cavité puisque les micro-surfaces sont modélisées séparément. La pondération des normales n'est donc pas dépendante du point de vue, ce qui est pourtant un effet important attendu sur une micro-surface réelle. C'est pourquoi le modèle à V-cavités n'incorpore pas cet effet de visibilité et modélise une surface qui ressemble plus à une normal map avec des micro-facettes plates sans existence géométrique qu'une height map, où les micro-facettes existent géométriquement.

Plus l'angle d'incidence est rasant, plus le profil de surface tend à produire cet effet "normal map". La conséquence pour la BRDF est que les distributions réfléchies sont globalement trop basses : sur une surface réelle les normales orientées dans la direction incidente ont une pondération plus importante dans la BRDF car leur aire projetée est plus importante et le lobe des directions réfléchies tend à être décalé vers le haut (figure 4.10). Cet effet de décalage du lobe n'est pas présent avec les normal maps parce que les micro-facettes n'existent pas géométriquement : elles ont toutes la même aire projetée.

La figure 4.11 montre les lobes des directions réfléchies par une micro-surface avec une distribution de normales de Beckmann isotrope et avec les profils de Smith et à V-cavités et avec une surface réelle créée

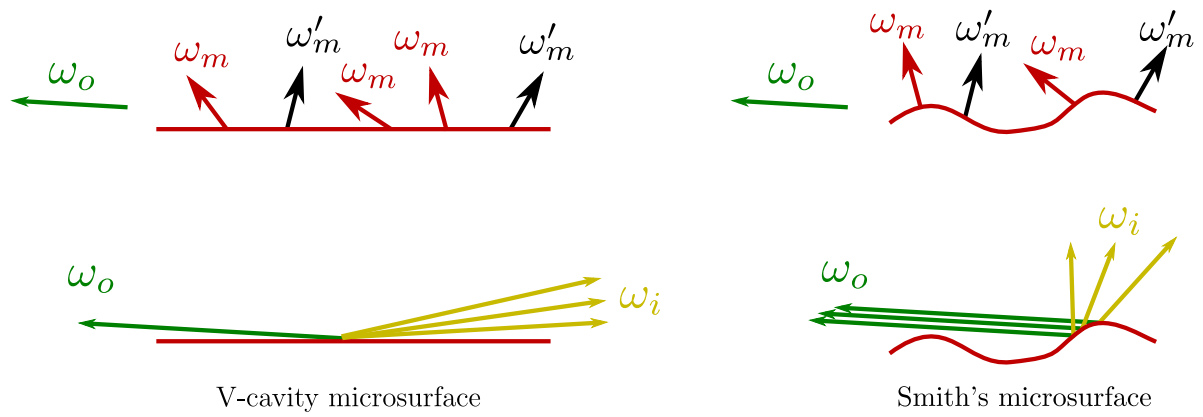


FIGURE 4.10 – (haut) Le modèle de micro-surface à V-cavités se comporte optiquement comme une normal map aux angles rasants : les normales non auto-masquées ont la même visibilité que la surface géométrique, comme si elles n'existaient pas géométriquement. (bas) Aux angles rasants, la distribution des directions réfléchies par la micro-surface est trop basse comparée à celle obtenue avec une surface physique.

de manière synthétique avec une fonction de bruit. On voit que, avec la fonction de masquage de Smith, le lobe réfléchi est décalé vers le haut lorsque la rugosité augmente. Pour les rugosités très importantes, on observe même un effet de “backscattering”. Cet effet, présent dans les données mesurées, était attendu, comme expliqué ci-dessus. Pourtant, il n'émerge pas avec le modèle à V-cavités.

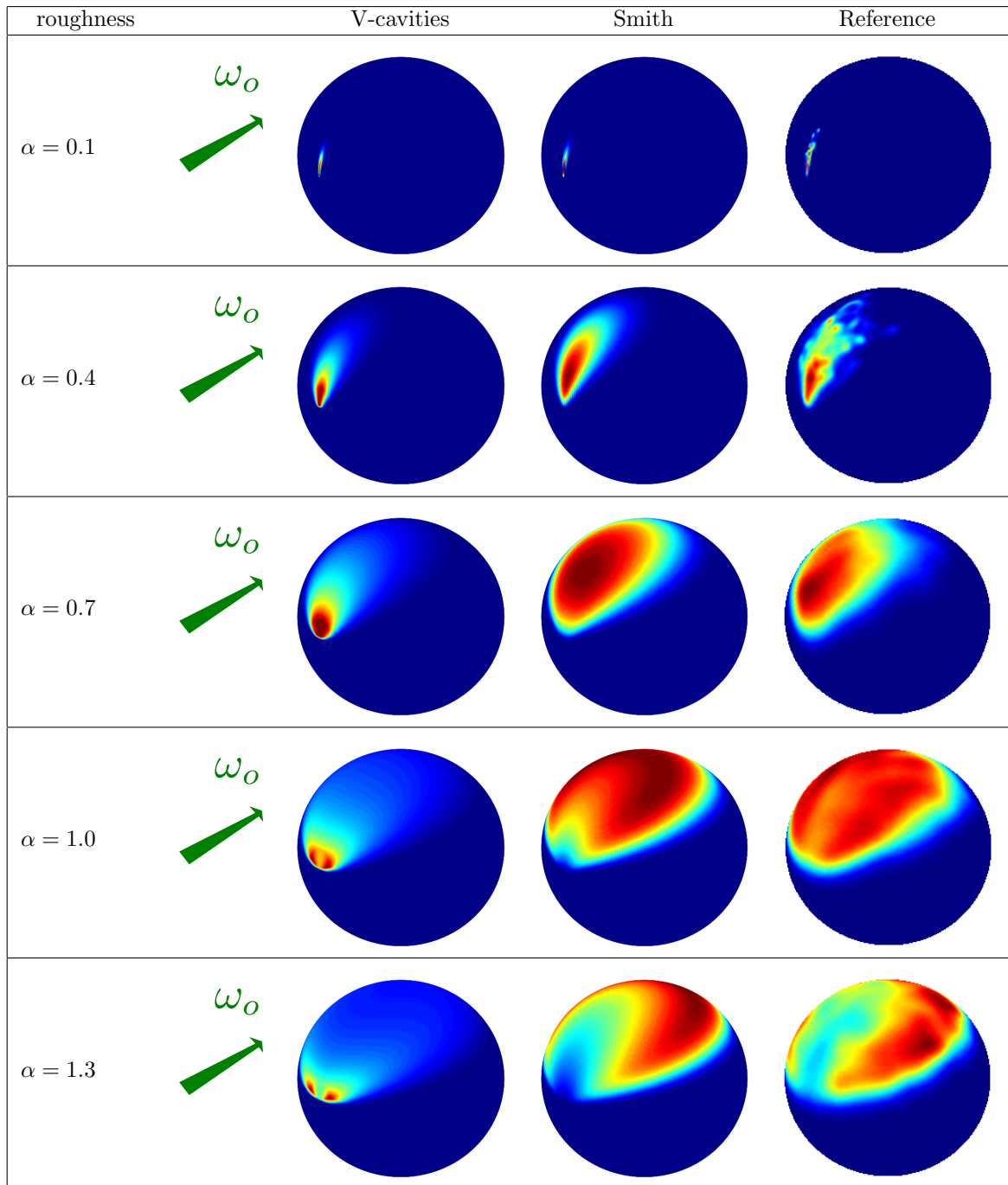


FIGURE 4.11 – (gauche et milieu) La distribution de directions réfléchies avec une distribution de normales de Beckmann isotrope et les fonctions de masquage et d'ombrage de Smith et à V-cavités ($\theta_o = 1.5$). (droite) Des données mesurées sur un échantillon synthétique généré avec une fonction de bruit procédurale à statistiques gaussiennes.

4.4.3 Les fonctions de masquages non physiques

Définition. Dans cette section, nous appelons *physically based* une fonction de masquage dont l'équation est dérivée à partir d'un modèle de micro-surface ou mesurée sur une micro-surface physique. Une fonction de masquage *physically based* vérifie toujours les équations (4.4), (4.8), (4.15) et (4.16). Réciproquement, nous appelons *non-physically based* une fonction de masquage qui ne satisfait pas ces équations, c'est-à-dire pour laquelle il n'existe aucune micro-surface à partir de laquelle la fonction de masquage pourrait être dérivée ou mesurée.

La fonction de masquage implicite. Pour simplifier l'expression de la BRDF à micro-facettes spéculaires donnée en équation (4.12), plusieurs modèles suppriment la fonction d'ombrage et de masquage et le dénominateur $|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|$, en faisant l'hypothèse qu'ils se simplifient [MHM⁺13]. La fonction de masquage et d'ombrage est donc définie de manière implicite par

$$G_2(\omega_o, \omega_i, \omega_m) = G_1(\omega_o, \omega_m) G_1(\omega_i, \omega_m), \quad (4.23)$$

où les fonctions de masquage et d'ombrage sont données par

$$G_1(\omega_o, \omega_m) = \chi^+(\omega_o \cdot \omega_m) \langle \omega_o, \omega_g \rangle, \quad (4.24)$$

$$G_1(\omega_i, \omega_m) = \chi^+(\omega_i \cdot \omega_m) \langle \omega_i, \omega_g \rangle. \quad (4.25)$$

Cette fonction de masquage est “plausible” car $G_1(\omega_o, \omega_m) = 1$ lorsque $\omega_o = \omega_g$ et décroît jusqu'à $G_1(\omega_o, \omega_m) = 0$ lorsque l'angle d'incidence tend vers $\frac{\pi}{2}$. Pourtant, cette fonction de masquage n'est pas *physically based*. En effet, elle ne vérifie pas l'équation (4.4) de la conservation de l'aire projetée, ce qui implique qu'il n'existe aucune micro-surface qui produit cette fonction de masquage.

La fonction de masquage de Schlick. Schlick propose une approximation de la fonction de masquage de Smith [Sch94], qui est souvent appelée “Schlick-Smith”. Cette fonction de masquage a deux problèmes.

Premièrement, le paramètre de rugosité m donné dans leur article est incohérent. Dans leur équation (18), il est défini comme étant l'écart-type des pentes de la micro-surface, c'est-à-dire $m = \sigma$, qui est le descripteur de rugosité le plus utilisé dans la littérature de physique et en particulier dans le papier original de Smith [Smi67]. Cela est cohérent avec la définition de leur variable h dans la même équation. En revanche, dans leur équation (20), m est utilisé comme le paramètre de rugosité de la distribution de Beckmann $m = \alpha = \sqrt{2}\sigma$, qui est l'écart-type des pentes multiplié par un facteur $\sqrt{2}$. De fait, la rugosité de la fonction de masquage ne correspond pas à celle utilisée dans la distribution de normales.

Mais ce premier problème, facile à résoudre, n'est pas le plus important : leur formule de la fonction de masquage de Smith, dans leur équation (18), est tout simplement fausse. Ils affirment “After several equivalences, the original expression of G can be written..”. Mais ces dérivations, qui ne sont pas explicitées, sont certainement fausses car le résultat obtenu ne correspond plus à la fonction originale, comme le montre la figure 4.12. Schlick a probablement réarrangé les équations données par He et al. [HTSG91] (équations (24) et (25) de cet article), qui contiennent une erreur : le terme exponentiel de la fonction Λ de Smith est manquant. L'approximation rationnelle qu'ils proposent par la suite approxime donc une fonction de référence qui est fausse.

De plus, à la fin de notre paragraphe “Les fonctions de Smith moyennées”, nous expliquons que la fonction de masquage de Smith à utiliser dans ce modèle de BRDF basé sur de l'optique géométrique est celle qui est moyennée sur les hauteurs uniquement. Mais Schlick utilise la forme moyennée sur les hauteurs et les normales, comme He et al., qui était appropriée dans leur modèle basé sur de l'optique ondulatoire, mais pas dans le sien.

En résumé, la fonction de masquage “Schlick-Smith” (la référence et l'approximation proposée) est tout simplement fausse, ne satisfait pas l'équation de la conservation de l'aire projetée et n'est en conséquence pas “physically-based”.

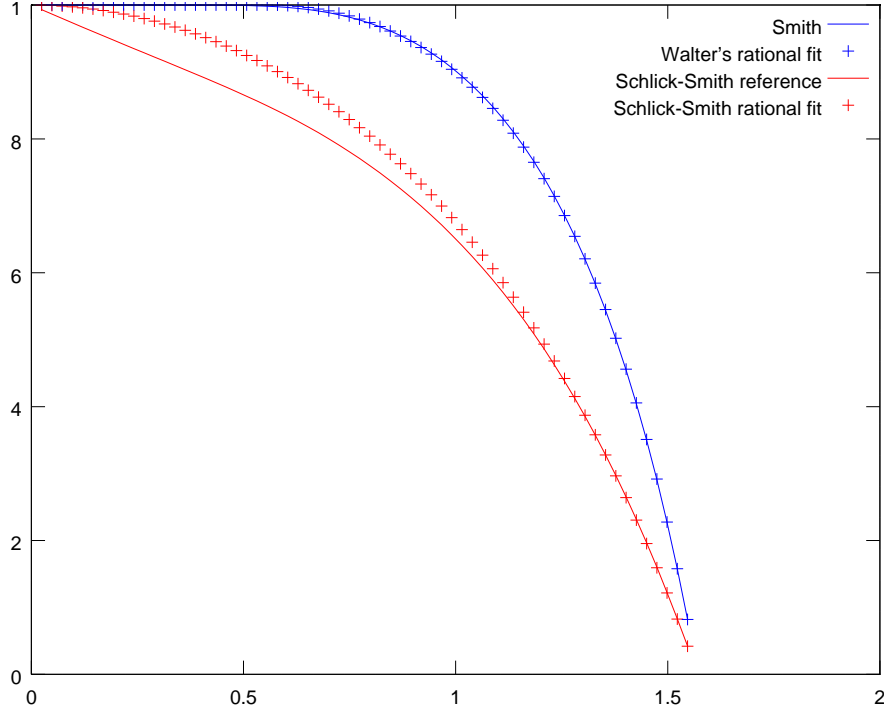


FIGURE 4.12 – La fonction de masquage originale de Smith, l’approximation rationnelle proposée par Walter, la fonction de référence (fausse) donnée par Schlick et l’approximation de Schlick (de la fonction fausse) en fonction de l’angle d’incidence θ_o et une rugosité fixée $\alpha = 1$.

La fonction de masquage de Kelemen. Dans le but de faire des calculs moins coûteux, Kelemen et al. [KSK01] proposent un moyen de remplacer la fonction de masquage et d’ombrage du modèle à V-cavités et les dénominateur du modèle à micro-facettes spéculaires donné dans l’équation (4.12) en approximant

$$\frac{G_2(\omega_o, \omega_i, \omega_h)}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} = \frac{1}{|\omega_o \cdot \omega_h| |\omega_i \cdot \omega_h|}, \quad (4.26)$$

ce qui revient à approximer les fonctions de masquage et d’ombrage séparément par

$$G_1(\omega_o, \omega_h) = \frac{|\omega_g \cdot \omega_o|}{|\omega_o \cdot \omega_h|}, \quad (4.27)$$

$$G_1(\omega_i, \omega_h) = \frac{|\omega_g \cdot \omega_i|}{|\omega_i \cdot \omega_h|}. \quad (4.28)$$

Cette fonction de masquage est proche de celle du modèle à V-cavité : le facteur $|\omega_g \cdot \omega_o|$ est l’aire projetée de la surface géométrique et la division par $|\omega_o \cdot \omega_h|$ permet de ne pas tenir compte de l’aire projetée de la micro-facette, c’est-à-dire que l’aire projetée de la micro-facette est remplacée par l’aire projetée de la surface géométrique. Cela tend à simuler la réflectance d’une micro-surface avec des micro-facettes plates comme sur une normal map.

En revanche, contrairement au modèle à V-cavités, la fonction de masquage de Kelemen ne satisfait pas la conservation de l’aire projetée de l’équation (4.4). Si elle constitue une bonne approximation au modèle à V-cavités, cette fonction de masquage ne peut pas être dérivée d’un modèle physique de micro-surface et n’est en conséquence pas “physically based”.

4.4.4 Résumé

Dans cette section, nous avons montré que :

- Les fonctions de masquage de Smith et à V-cavités sont toutes les deux exactes, mais supposent différents profils de micro-surface.
- Les deux satisfont l'équation (4.4) de la conservation de l'aire projetée.
- Les deux sont le coefficient de normalisation de la distribution de normales visibles de l'équation (4.8).
- Les deux satisfont le *Weak White Furnace test* donné par les équations (4.15) et (4.16).
- Au contraire, les fonctions de masquage implicite, de Schlick et de Kelemen ne satisfont pas les équations (4.4), (4.8), (4.15) et (4.16). Elles ne sont donc pas *physically based* dans le sens où il n'existe pas de profils de micro-surface desquels elles peuvent être dérivées.

fonction de masquage	Equations (4.4), (4.8), (4.15) and (4.16)	physically based
Smith	✓	✓
V-cavités	✓	✓
Implicite	✗	✗
Schlick-Smith	✗	✗
Kelemen	✗	✗

TABLE 4.3 – Propriétés des différentes fonctions de masquage.

La fonction de masquage de Smith. Une croyance typique est : “*La fonction de masquage de Smith est une bonne approximation parce qu'elle dépend de la distribution de normales.*”

Nous avons vu que cette réponse est correcte mais la raison donnée est mauvaise :

- Choisir la fonction de masquage de Smith implique de faire le choix de supposer une micro-surface sur laquelle il y a indépendance des normales et de la fonction de masquage.
- Sous cette hypothèse, la fonction de masquage est complètement déterminée ; sa forme exacte peut être dérivée et elle est la fonction de Smith généralisée.

Nous insistons ici sur le fait que la raison pour laquelle on choisit la fonction de masquage de Smith n'est pas que c'est une approximation plausible paramétrée par la distribution de normales. La vraie raison pour choisir la fonction de masquage de Smith est que c'est la seule et unique fonction de masquage sous l'hypothèse du modèle : indépendance normal/masquage. Le fait qu'elle soit physiquement plausible et paramétrée par la distribution de normales ne constitue pas en soi la raison pour la choisir, mais sont des effets de bord de faire le bon choix.

La fonction de masquage à V-cavités. Une autre question récurrente est : “*La fonction de masquage à V-cavités est-elle fausse ?*”

La réponse typique à cette question est la suivante : “*Elle est fausse car elle ne dépend pas de la distribution de normales.*”

Nous avons vu que cette réponse est fausse pour les raisons suivantes :

- La fonction de masquage à V-cavités ne dépend pas, par normale, de la distribution de normales.

- Par contre, la moyenne de la fonction de masquage à V-cavités dépend de la distribution de normales. C'est parce que la fonction de masquage et les normales ne sont pas indépendantes, comme dans le modèle de Smith. Plus la rugosité augmente, plus la fonction de masquage moyenne augmente.
- La fonction de masquage à V-cavités peut être utilisée avec n'importe quelle distribution de normales symétrique et garantit une normalisation correcte.
- Par contre, le profil de surface a une réponse plus proche de celle obtenue avec une normal map qu'avec une height map, surtout aux angles rasants. C'est la raison pour laquelle on la qualifie de non physiquement réaliste.
- La conséquence est qu'aux angles rasants, avec des rugosités importantes, le lobe réfléchi de la BRDF est très bas comparé à ce qu'on attend d'un matériau réaliste.

En résumé, il n'y a pas de réponse définitive à la question de choisir le profil de micro-surface de Smith ou à V-cavités, car le choix est arbitraire (ils modélisent différentes micro-surfaces) et les deux sont exacts sous leurs hypothèses respectives. En pratique, la fonction à V-cavité a l'avantage d'être moins coûteuse et surtout d'être générique puisqu'elle fonctionne avec n'importe quelle distribution de normales (symétrique). D'un autre côté, la fonction de Smith est plus coûteuse et demande des dérivations spécifiques, mais le profil de micro-surface qu'elle suppose est plus proche de celui qu'une surface stochastique réelle aurait. Le choix entre Smith et les V-cavités est donc une question de compromis entre le réalisme et les performances.

→ A partir de maintenant, nous faisons le choix d'utiliser la fonction de masquage de Smith, car nous souhaitons justement résoudre le problème de filtrage de l'apparence des height maps, et non celui du filtrage de l'apparence des normal maps.

4.5 Invariance d'étirement de la fonction de masquage

Dans cette section, nous étudions la propriété d'invariance de la fonction de masquage et de la distribution des pentes de la micro-surface lorsque la configuration géométrique est étirée. Nous utilisons cette connaissance pour dériver les fonctions de masquage de Smith pour des distributions de normales anisotropes.

4.5.1 Invariance de la probabilité de masquage

La figure 4.13 montre l'effet que produit un étirement d'une configuration 1D avec le masquage d'une micro-surface pour une certaine direction sortante ω_o . Étirer la configuration revient intuitivement à étirer l'image, c'est-à-dire qu'une des dimensions de l'image est multipliée par un facteur constant. Cette opération ne change pas la topologie de la configuration : après l'étirement, les rayons occlus sont toujours occlus et les rayons non occlus ne le sont toujours pas. Il y a donc une invariance topologique de la configuration sous étirement : la probabilité de masquage est invariante aux étirements de la configuration, où toutes les pentes de la configuration sont étirées en même temps. Cela inclut les pentes de la micro-surface mais aussi la pente associée à la direction d'observation. Elles sont toutes multipliées par l'inverse du facteur d'étirement. La distribution de pente quant à elle, est étirée par l'inverse du facteur d'étirement : plus les pentes sont faibles, plus elles sont distribuées proche de 0 dans leur histogramme.

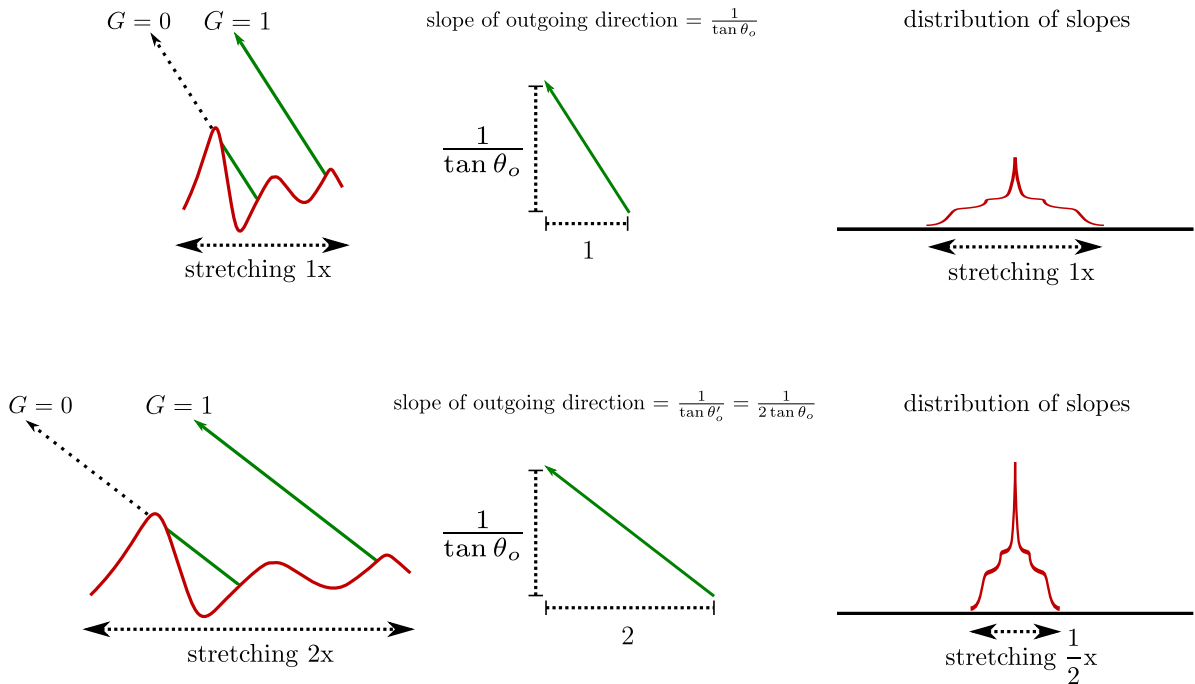


FIGURE 4.13 – Étirer une configuration 1D par un facteur 2 ne change pas la probabilité de masquage, mais toutes les pentes de la configuration sont divisées par 2. Cela inclut les pentes de la micro-surface ainsi que la pente associée à la direction d'observation.

4.5.2 La distribution des pentes

Si la micro-surface est une height map, la distribution des hauteurs de la micro-surface est souvent notée $P^1(h)$. Les pentes de la surface sont les gradients de la fonction de hauteur $(x_{\tilde{m}}, y_{\tilde{m}}) = \nabla h$: elles mesurent comment la hauteur de la micro-surface varie spatialement. La distribution des pentes est notée

$P^{22}(x_{\tilde{m}}, y_{\tilde{m}})$ où les pentes sont telles que :

$$\tilde{m} = (x_{\tilde{m}}, y_{\tilde{m}}) = \left(-\frac{x_m}{z_m}, -\frac{y_m}{z_m} \right) = -\tan \theta_m (\cos \phi_m, \sin \phi_m), \quad (4.29)$$

où la normale est donnée par $\omega_m = (x_m, y_m, z_m)$, et réciproquement :

$$\omega_m = \frac{(-x_{\tilde{m}}, -y_{\tilde{m}}, 1)}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}}.$$

La distribution des pentes est normalisée :

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dx_{\tilde{m}} dy_{\tilde{m}} = 1,$$

et la distribution de normales est définie par :

$$D(\omega_m) = \frac{P^{22}(x_{\tilde{m}}, y_{\tilde{m}})}{\cos^4 \theta_m}.$$

Lorsque le paramètre de rugosité doit être explicite, nous notons $D(\omega_m, \alpha)$ et $P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha)$ pour des distributions isotropes et $D(\omega_m, \alpha_x, \alpha_y)$ et $P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y)$ pour des distributions anisotropes.

4.5.3 Distributions de pentes isotropes à forme invariante

Invariance de forme. Plusieurs distributions de pentes isotropes P^{22} dépendent d'un paramètre de rugosité α , où changer α est équivalent à étirer la distribution des pentes sans changer sa forme. C'est le cas lorsque la distribution des pentes ne dépend que du rapport $\frac{\tan \theta_m}{\alpha} = \frac{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2}}{\alpha}$ entre l'amplitude de la pente $\tan \theta_m = \sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2}$ d'une normale d'angle θ_m et le paramètre de rugosité α :

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha) = \frac{1}{\alpha^2} f \left(\frac{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2}}{\alpha} \right) = \frac{1}{\alpha^2} f \left(\frac{\tan \theta_m}{\alpha} \right), \quad (4.30)$$

où f est une fonction 1D qui définit la forme de la distribution. Ces distributions de pentes sont *invariantes de forme*, car elles ont toujours la même forme f et ne sont que étirées et renormalisées par le paramètre de rugosité :

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha) = \frac{1}{\lambda^2} P^{22}\left(\frac{x_{\tilde{m}}}{\lambda}, \frac{y_{\tilde{m}}}{\lambda}, \frac{\alpha}{\lambda}\right), \quad \text{pour tout } \lambda > 0.$$

Comme le montre la figure 4.13, avec des distributions de pentes invariantes de forme, étirer la configuration équivaut à multiplier la rugosité α et la pente de la direction d'observation par le même facteur. Cela implique que la fonction de masquage ne dépend que du rapport $a = \frac{1}{\alpha \tan \theta_o}$, où $\frac{1}{\tan \theta_o}$ est la pente associée à la direction d'observation². Les distributions de Beckmann et GGX sont invariantes de forme et c'est pour cette raison que leurs fonctions Λ , définies par la fonction de masquage, ne dépendent que du rapport a .

La distribution de Beckmann.

$$\begin{aligned} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) &= \frac{1}{\pi \alpha^2} \exp \left(-\frac{x_{\tilde{m}}^2 + y_{\tilde{m}}^2}{\alpha^2} \right), \\ D(\omega_m) &= \frac{\chi^+(\omega_m \cdot \omega_g)}{\pi \alpha^2 \cos^4 \theta_m} \exp \left(-\frac{\tan^2 \theta_m}{\alpha^2} \right), \\ \Lambda(\omega_o) &= \frac{\operatorname{erf}(a) - 1}{2} + \frac{1}{2a\sqrt{\pi}} \exp(-a^2). \end{aligned}$$

2. Pour un angle d'incidence θ donné, la pente associée à cette direction est $\frac{1}{\tan \theta_o}$ et ne doit pas être confondue avec la pente $\tan \theta$ d'une micro-facette orthogonale à cette direction.

où $a = \frac{1}{\alpha \tan \theta_o}$. Walter et al. [WMLT07] proposent une approximation pour $G_1(\omega_o) = \frac{1}{1+\Lambda(\omega_o)}$, qui peut-être utilisée pour approximer $\Lambda(\omega_o)$ (via $\Lambda(\omega_o) = \frac{1-G_1(\omega_o)}{G_1(\omega_o)}$) :

$$\Lambda(\omega_o) \approx \begin{cases} \frac{1-1.259a+0.396a^2}{3.535a+2.181a^2} & \text{si } a < 1.6 \\ 0 & \text{sinon.} \end{cases}$$

La distribution GGX.

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) = \frac{1}{\pi \alpha^2 \left(1 + \frac{x_{\tilde{m}}^2 + y_{\tilde{m}}^2}{\alpha^2}\right)^2},$$

$$D(\omega_m) = \frac{\chi^+(\omega_m \cdot \omega_g)}{\pi \alpha^2 \cos^4 \theta_m \left(1 + \frac{\tan^2 \theta_m}{\alpha^2}\right)^2},$$

$$\Lambda(\omega_o) = \frac{-1 + \sqrt{1 + \frac{1}{a^2}}}{2}.$$

où $a = \frac{1}{\alpha \tan \theta_o}$.

Les distributions à forme variante. A noter que toutes les distributions n'ont pas cette propriété. En particulier, la distribution de Phong n'est pas invariante de forme car elle ne peut pas être exprimée sous la forme de l'équation (4.30) : lorsque la rugosité change, la forme de la distribution de Phong change.

4.5.4 Distributions de pentes anisotropes à forme invariante

Invariance de forme. Les mêmes distributions invariantes de forme peuvent être anisotropes si la forme est étirée avec des facteurs dépendants de la direction. Les pentes sont pondérées différemment dans chaque direction et l'équation (4.30) devient :

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y) = \frac{1}{\alpha_x \alpha_y} f \left(\sqrt{\frac{x_{\tilde{m}}^2}{\alpha_x^2} + \frac{y_{\tilde{m}}^2}{\alpha_y^2}} \right) = \frac{1}{\alpha_x \alpha_y} f \left(\tan \theta_m \sqrt{\frac{\cos^2 \phi_m}{\alpha_x^2} + \frac{\sin^2 \phi_m}{\alpha_y^2}} \right), \quad (4.31)$$

où $(\tan \theta_m \cos \phi_m, \tan \theta_m \sin \phi_m) = (x_{\tilde{m}}, y_{\tilde{m}})$ sont les pentes et α_x and α_y sont les coefficient d'étirement de la distribution sur les axes x et y respectivement. La propriété d'invariance de forme est donnée par :

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y) = \frac{1}{\lambda_x \lambda_y} P^{22}\left(\frac{x_{\tilde{m}}}{\lambda_x}, \frac{y_{\tilde{m}}}{\lambda_y}, \frac{\alpha_x}{\lambda_x}, \frac{\alpha_y}{\lambda_y}\right), \quad \text{for any } \lambda_x, \lambda_y > 0. \quad (4.32)$$

Dérivation de la fonction de masquage. La figure 4.14 montre comment des distributions isotropes invariantes de forme peuvent être transformées en distributions anisotropes en étirant la surface. Réciproquement, toutes les configurations avec des distributions anisotropes peuvent être transformées en des configurations avec des distributions isotropes.

Nous utilisons cette propriété pour dériver la fonction de masquage pour des distributions anisotropes. Nous commençons avec une configuration avec une distribution anisotrope invariante de forme de paramètres α_x et α_y et une direction d'observation $\omega_o = (x_o, y_o, z_o)$. En étirant la surface le long de l'axe x d'un facteur $\frac{\alpha_x}{\alpha_y}$ la rugosité de la surface devient :

$$\begin{aligned} \alpha'_x &= \alpha_x \frac{\alpha_y}{\alpha_x} \\ &= \alpha_y, \\ \alpha'_y &= \alpha_y. \end{aligned}$$

La surface est maintenant isotrope de rugosité α_y et la direction d'observation et sa pente après étirement sont données par :

$$\begin{aligned}\omega'_o &= \left(\frac{\alpha_x}{\alpha_y} x_o, y_o, z_o \right) \\ &= \left(\frac{\alpha_x}{\alpha_y} \cos \phi_o \sin \theta_o, \sin \phi_o \sin \theta_o, \cos \theta_o \right), \\ \frac{1}{\tan \theta'_o} &= \frac{z_o}{\sqrt{\frac{\alpha_x^2}{\alpha_y^2} x_o^2 + y_o^2}} \\ &= \frac{1}{\sqrt{\frac{\alpha_x^2}{\alpha_y^2} \cos^2 \phi_o + \sin^2 \phi_o} \tan \theta_o}.\end{aligned}$$

La fonction de masquage d'une distribution isotrope dépend uniquement du rapport $a = \frac{1}{\alpha \tan \theta_o}$ et comme $\alpha = \alpha_y$ le rapport de la surface étiré est :

$$\begin{aligned}a' &= \frac{1}{\alpha_y \tan \theta'_o} \\ &= \frac{1}{\alpha_y \sqrt{\cos^2 \phi_o \frac{\alpha_x^2}{\alpha_y^2} + \sin^2 \phi_o} \tan \theta_o} \\ &= \frac{1}{\sqrt{\cos^2 \phi_o \alpha_x^2 + \sin^2 \phi_o \alpha_y^2} \tan \theta_o} \\ &= \frac{1}{\alpha_o \tan \theta_o},\end{aligned}$$

où

$$\alpha_o = \sqrt{\cos^2 \phi_o \alpha_x^2 + \sin^2 \phi_o \alpha_y^2}, \quad (4.33)$$

est la *rugosité projetée dans la direction d'observation*. Cela montre que la fonction de masquage associée à des distributions de pentes anisotropes et invariantes de forme est tout simplement la fonction de masquage associée aux distributions isotropes de la même classe. Celle-ci prend alors en paramètre la rugosité de la surface anisotrope projetée dans la direction d'observation. Nous utilisons cette propriété pour dériver la fonction de masquage pour les distributions de Beckmann et GGX anisotropes.

La distribution de Beckmann anisotrope.

$$\begin{aligned}P^{22}(x_{\bar{m}}, y_{\bar{m}}) &= \frac{1}{\pi \alpha_x \alpha_y} \exp \left(-\frac{x_{\bar{m}}^2}{\alpha_x^2} - \frac{y_{\bar{m}}^2}{\alpha_y^2} \right), \\ D(\omega_m) &= \frac{\chi^+(\omega_m \cdot \omega_g)}{\pi \alpha_x \alpha_y \cos^4 \theta_m} \exp \left(-\tan^2 \theta_m \left(\frac{\cos^2 \phi_m}{\alpha_x^2} + \frac{\sin^2 \phi_m}{\alpha_y^2} \right) \right), \\ \Lambda(\omega_o) &= \frac{\operatorname{erf}(a) - 1}{2} + \frac{1}{2 a \sqrt{\pi}} \exp(-a^2).\end{aligned}$$

où $a = \frac{1}{\alpha_o \tan \theta_o}$ et α_o est défini dans l'équation 4.33. L'approximation de Λ pour la distribution de Beckmann isotrope peut aussi être utilisée ici.

La distribution GGX anisotrope.

$$P^{22}(x_{\bar{m}}, y_{\bar{m}}) = \frac{1}{\pi \alpha_x \alpha_y \left(1 + \frac{x_{\bar{m}}^2}{\alpha_x^2} + \frac{y_{\bar{m}}^2}{\alpha_y^2}\right)^2},$$

$$D(\omega_m) = \frac{\chi^+(\omega_m \cdot \omega_g)}{\pi \alpha_x \alpha_y \cos^4 \theta_m \left(1 + \tan^2 \theta_m \left(\frac{\cos^2 \phi_m}{\alpha_x^2} + \frac{\sin^2 \phi_m}{\alpha_y^2}\right)\right)^2},$$

$$\Lambda(\omega_o) = \frac{-1 + \sqrt{1 + \frac{1}{a^2}}}{2}.$$

où $a = \frac{1}{\alpha_o \tan \theta_o}$ et α_o est défini dans l'équation 4.33.

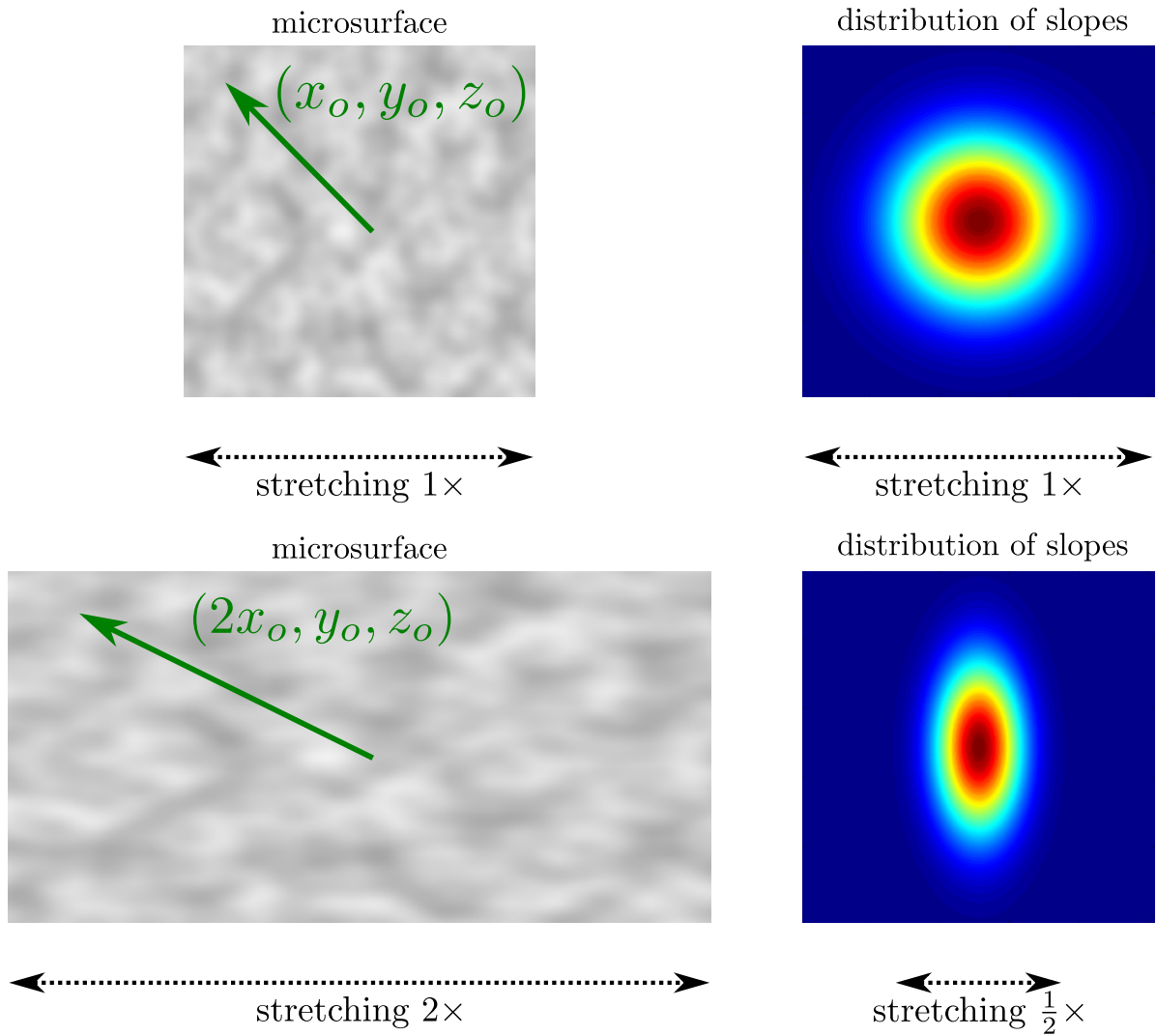


FIGURE 4.14 – Etirer une configuration 2D par un facteur 2. Les coordonnées de la direction d'observation sont étirées aussi.

4.5.5 Plus de généralisation

Des distributions invariantes de formes arbitraires. Une propriété importante des distributions invariantes de forme est que toute l'information requise pour la fonction de masquage est contenue dans la même fonction Λ pour n'importe quelle rugosité ou anisotropie. Donc, si Λ est connue, elle peut être utilisée pour une classe entière de distributions paramétrique de pentes avec une rugosité et une anisotropie variantes. Il est facile de se créer sa propre distribution de normales en choisissant une fonction 1D f arbitraire et en posant :

$$D(\omega_m) = \frac{c}{\alpha_x \alpha_y \cos^4 \theta_m} f \left(\tan^2 \theta_m \left(\frac{\cos^2 \phi_m}{\alpha_x^2} + \frac{\sin^2 \phi_m}{\alpha_y^2} \right) \right).$$

où c serait le coefficient constant de normalisation de la distribution. La fonction 1D $\Lambda(\frac{1}{\alpha_o \tan \theta_o})$ associée peut être pré-calculée numériquement et stockée dans une table ou approximée avec un polynôme rationnel, comme l'ont fait Walter et al. [WMLT07] pour la distribution de Beckmann.

Étirement non aligné sur les axes. L'opération d'étirement n'est pas nécessairement alignée sur les axes x et y . L'opération d'étirement dans l'espace des pentes peut être formulée de manière générale avec une quadrique. Soit Q une matrice symétrique définie positive³ :

$$Q^{-1} = \begin{bmatrix} \alpha_x^2 & r_{xy} \alpha_x \alpha_y \\ r_{xy} \alpha_x \alpha_y & \alpha_y^2 \end{bmatrix},$$

et r_{xy} est le coefficient de corrélation de l'étirement sur les axes x et y . La quadrique Q définit un produit scalaire et une norme dans l'espace euclidien 2D des pentes :

$$\begin{aligned} \|\tilde{m}\| &= \sqrt{\langle \tilde{m}, \tilde{m} \rangle} \\ &= \sqrt{\tilde{m}^T Q \tilde{m}}, \end{aligned}$$

où le vecteur 2D $\tilde{m} = -\tan \theta_m (\cos \phi_m, \sin \phi_m)$ défini dans l'équation (4.29) est la pente associée à la normale ω_m . La norme du vecteur de pente $\|\tilde{m}\|$ décrit l'étirement qui se produit dans l'espace des pentes et elle est l'argument de la distribution D . Nous avons donné la formule pour la norme et la rugosité projetée dans la direction d'observation dans le cas où $r_{xy} = 0$:

$$\begin{aligned} \|\tilde{m}\|^2 &= \tan^2 \theta_m (\cos \phi_m, \sin \phi_m)^T Q \tan \theta_m (\sin \phi_m, \cos \phi_m) \\ &= \tan^2 \theta_m \left(\frac{\cos^2 \phi_m}{\alpha_x^2} + \frac{\sin^2 \phi_m}{\alpha_y^2} \right), \\ \alpha_o^2 &= \cos^2 \phi_o \alpha_x^2 + \sin^2 \phi_o \alpha_y^2. \end{aligned}$$

Dans le cas général $r_{xy} \neq 0$ et nous avons :

$$\begin{aligned} \|\tilde{m}\| &= \tan \theta_m (\cos \phi_m, \sin \phi_m)^T Q \tan \theta_m (\sin \phi_m, \cos \phi_m) \\ &= \tan^2 \theta_m \left(\frac{\cos^2 \phi_m \alpha_y^2 + \sin^2 \phi_m \alpha_x^2 - 2 \cos \phi_m \sin \phi_m r_{xy} \alpha_x \alpha_y}{\alpha_x^2 \alpha_y^2 - r_{xy}^2 \alpha_x^2 \alpha_y^2} \right), \\ \alpha_o^2 &= \cos^2 \phi_o \alpha_x^2 + \sin^2 \phi_o \alpha_y^2 + 2 \cos \phi_o \sin \phi_o r_{xy} \alpha_x \alpha_y. \end{aligned}$$

Par exemple, dans *LEAN Mapping* [OB10], la représentation utilisée pour décrire la distribution des pentes est une matrice de covariance corrélée et utilisée dans une distribution de Beckmann. Nous discuterons cette représentation en détail dans le chapitre 8. A noter qu'un coefficient de corrélation est à valeur dans $r_{xy} \in [-1, 1]$ et affecte la constante de normalisation de la distribution.

3. Dans le cas particulier de la distribution de Beckmann, $\Sigma = \frac{1}{2} Q^{-1}$ est la matrice de covariance de la distribution gaussienne de pentes.

Chapitre 5

Théorie multi-échelles des micro-facettes

La théorie des micro-facettes que nous avons vue dans le chapitre précédent est une théorie *bi-échelles* qui ne fait intervenir que l'échelle macroscopique, que nous appelons la surface géométrique, et l'échelle microscopique, que nous appelons la micro-surface. Cette formulation de la théorie des micro-facettes ne permet pas son utilisation dans un framework *multi-échelles* où la transition entre les échelles est continue. En effet, il manque l'échelle *mésoscopique*, illustrée dans la figure 5.1, qui permet cette transition. Dans ce chapitre, nous commençons par introduire cette échelle dans la théorie des micro-facettes. Dans la section 5.1 nous allons voir que l'échelle mésoscopique est modélisable par un *décentrage* de la distribution des pentes de la micro-surface. Ce décentrage a pour conséquence directe de changer la valeur de l'aire projetée de la micro-surface dans la direction d'observation, valeur qui intervient dans tous les résultats dérivés dans le chapitre 4. Pour mettre au point un modèle qui incorpore correctement l'échelle mésoscopique, nous étendons dans la section 5.2, les résultats dérivés dans le chapitre 4 à des distributions de pentes *décentrées*.

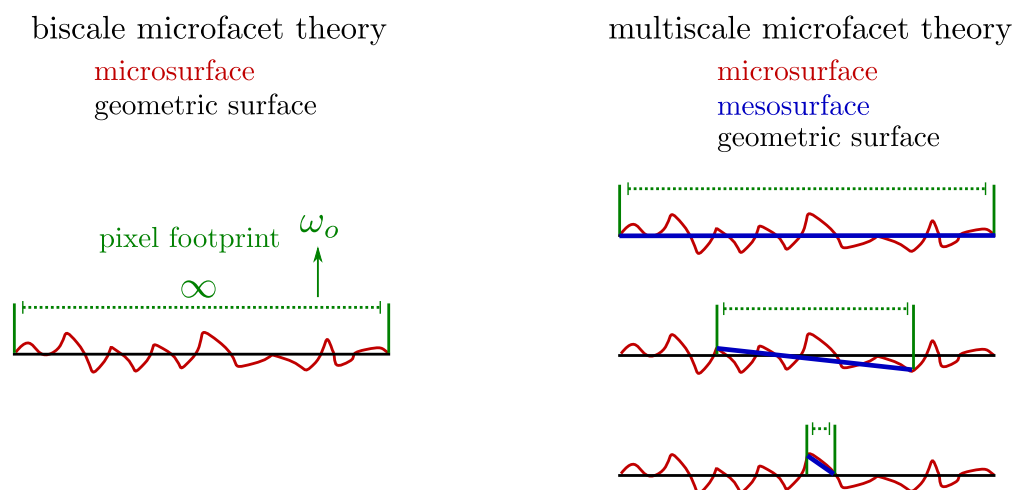


FIGURE 5.1 – (gauche) La théorie standard des micro-facettes est bi-échelles et suppose que la micro-surface soit très petite par rapport à l'échelle d'observation. (droite) Cette hypothèse ne tient plus dans une théorie multi-échelles où l'échelle d'observation est ajustée en fonction de la distance de l'observateur.

5.1 Introduction de l'échelle mésoscopique

5.1.1 Motivation

La théorie des micro-facettes que nous avons vue jusqu'à présent est une théorie bi-échelles dans laquelle on distingue l'échelle macroscopique, la surface géométrique, et l'échelle microscopique, la micro-surface. Les deux échelles sont fixées à l'avance : la micro-surface est très petite par rapport à l'échelle d'observation et n'est pas observable directement, mais seulement à travers ses effets photométriques. Dans un modèle multi-échelles, nous souhaitons au contraire pouvoir ajuster de manière dynamique l'échelle d'observation, comme le montre la figure 5.1. Cet ajustement signifie que l'hypothèse admettant que la micro-surface soit très petite par rapport à l'échelle d'observation ne tient plus. On observe alors l'apparition d'une nouvelle échelle : l'échelle mésoscopique.

5.1.2 Définition de l'échelle mésoscopique

Dans la figure 5.1, la *méso-surface* correspond au plan moyen (en bleu) de la micro-surface observable, c'est-à-dire la *pente moyenne* de la micro-surface observable, qui s'écrit :

$$(\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x_{\tilde{m}}, y_{\tilde{m}}) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dx_{\tilde{m}} dy_{\tilde{m}}.$$

En théorie bi-échelles des micro-facettes, on suppose que l'échelle d'observation est très grande par rapport aux variations de la micro-surface. Dans ce cas, la *loi des grands nombres* s'applique et statistiquement, le plan moyen de la micro-surface est le plan de la surface géométrique dont la pente est nulle :

$$(\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}) = (0, 0),$$

et la méso-surface est confondue avec la macro-surface. C'est pour cette raison que la méso-surface n'est jamais considérée, ni même définie, en théorie bi-échelles des micro-facettes. En revanche, dans la théorie multi-échelles des micro-facettes que nous cherchons à mettre au point, l'échelle d'observation est ajustable. Lorsqu'elle n'est pas beaucoup plus grande que la taille de la micro-surface, la loi des grands nombres ne s'applique plus et dans le cas général :

$$(\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}) \neq (0, 0),$$

la méso-surface n'est pas confondue avec la macro-surface. On peut voir la pente de la méso-surface comme étant l'espérance de la distribution des pentes P^{22} :

$$(\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}) = \mathbb{E}[(x_{\tilde{m}}, y_{\tilde{m}})].$$

Lorsqu'une distribution de pentes a une espérance (une moyenne) non nulle, nous dirons qu'elle est *décentrée*. Les distributions utilisées dans la littérature pour définir les BRDFs sont toujours centrées, puisqu'elles sont utilisées dans le modèle bi-échelles uniquement. Dans le modèle multi-échelles que nous souhaitons définir, nous devons formuler correctement la BRDF avec des distributions décentrées. Dans un premier temps, le décentrage se traduit concrètement par une simple translation des pentes dans P^{22} . Voici comment il se traduit concrètement pour le cas des distributions de Beckmann et GGX introduites dans le chapitre 4.

La distribution de Beckmann décentrée.

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}) = \frac{1}{\pi \alpha^2} \exp\left(-\frac{(x_{\tilde{m}} - \bar{x}_{\tilde{m}})^2 + (y_{\tilde{m}} - \bar{y}_{\tilde{m}})^2}{\alpha^2}\right).$$

La distribution GGX décentrée.

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}) = \frac{1}{\pi \alpha^2 \left(1 + \frac{(x_{\tilde{m}} - \bar{x}_{\tilde{m}})^2 + (y_{\tilde{m}} - \bar{y}_{\tilde{m}})^2}{\alpha^2}\right)^2}.$$

5.1.3 Les approches précédentes

Certaines approches précédentes utilisent la même description multi-échelles de la surface que la nôtre. En particulier, Eric Bruneton et Fabrice Neyret utilisent une gaussienne décentrée pour représenter la distribution des pentes P^{22} de la surface multi-échelles de la mer [BNH10]. Nous avons constaté, en visite chez Weta Digital, qu'ils utilisent aussi cette représentation.

Problème. Ils se trouvent alors confrontés au problème de devoir choisir un modèle de BRDF paramétré par une distribution de pentes décentrée. Mais les équations des micro-facettes que nous avons vues précédemment supposent toujours que les distributions sont centrées, c'est-à-dire que la méso-surface est confondue avec la surface géométrique, et ne s'appliquent pas dans le cas décentré. Autrement dit, leur problème est qu'ils ne savent pas quelles sont les équations du modèle décentré qu'ils doivent calculer. Que faire alors ?

Pour se tirer d'affaire et se ramener au modèle bi-échelles centré dont les équations sont connues, ils redéfinissent localement la surface géométrique comme étant la méso-surface. Intuitivement, ils gardent les paramètres de rugosité de la micro-surface, mais font subir une rotation au repère dans lequel ils sont définis. Mais cette technique est mal posée car la micro-surface décentrée n'est pas la micro-surface tournée et alignée autour de la méso-normale, comme le montre la figure 5.3. Pour la même raison, les statistiques de rugosité de la micro-surface ne sont pas conservées lorsqu'on change de repère, comme le montre la figure 5.3. Cette technique est donc mal définie et produit un résultat faux. En particulier, on peut voir dans la figure 5.3 que la technique qui consiste à faire une rotation du repère produit une rugosité trop importante (le lobe décentré est bien moins large que le lobe centré tourné) et le décentrage produit une certaine anisotropie qui n'est pas présente dans la distribution tournée. Enfin, la distribution de normales tournée a des normales qui ne sont plus dans l'hémisphère supérieure, ce qui est incompatible avec le fait d'être sur une height map. Cela peut avoir pour effet de réfléchir des directions d'éclairage qui ne devraient normalement pas être visibles.

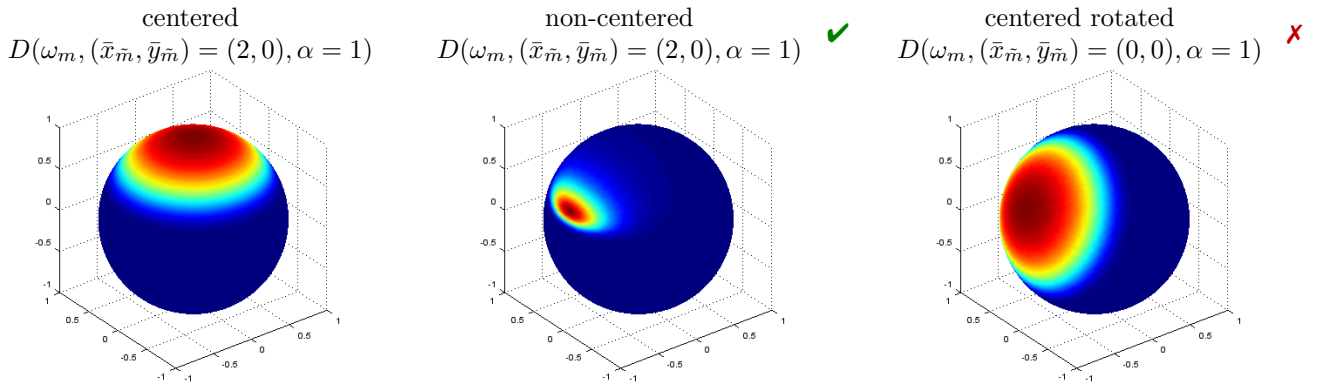
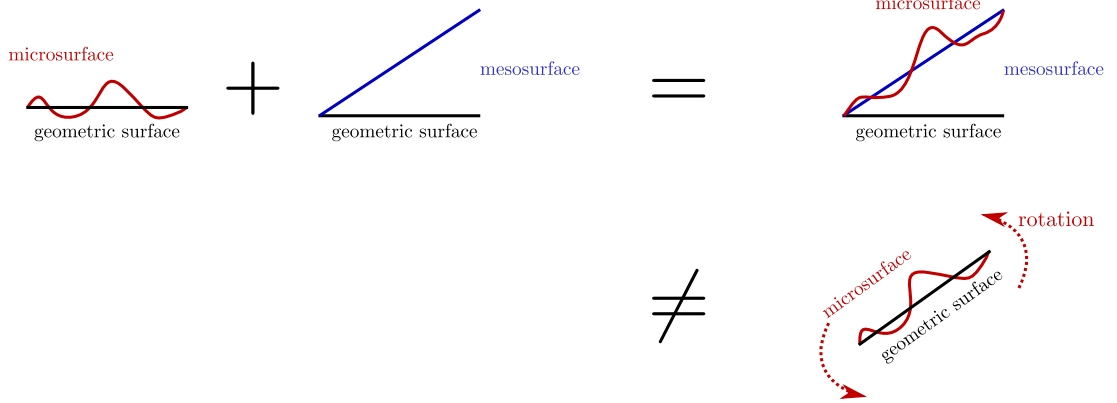


FIGURE 5.2 – *Distribution de Beckmann décentrée. La distribution de normales de la micro-surface décentrée n'est pas la rotation de la distribution de normales d'une micro-surface centrée.*

Notre solution. Il n'est pas possible de se ramener au modèle bi-échelles en faisant une simple rotation du repère. Pour pouvoir utiliser les distributions de pentes décentrées indispensables au modèle multi-échelles, nous allons redériver les équations des micro-facettes pour une micro-surface décentrée.

spatial interpretation



statistical interpretation

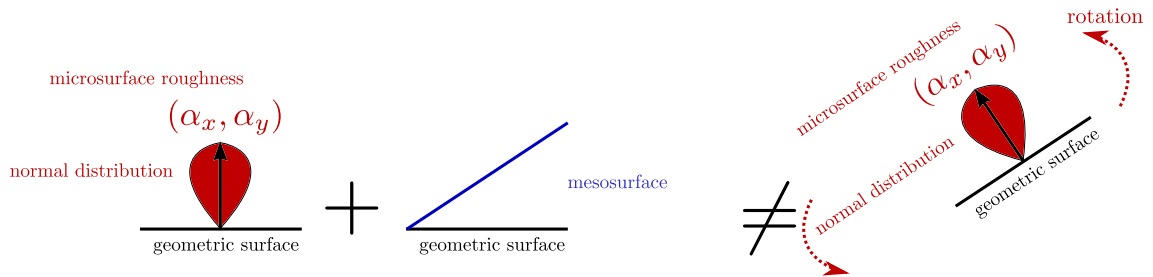


FIGURE 5.3 – La micro-surface décentrée n'est pas la micro-surface tournée et alignée autour de la méso-normale. Les statistiques de rugosité de la micro-surface ne sont pas conservées lorsqu'on change de repère. Il n'est donc pas possible de se ramener au modèle bi-échelles où la méso-surface et la surface géométrique sont confondues en faisant une rotation du repère géométrique.

5.2 Redériver les résultats précédents pour des distributions décentrées

Dans cette section, nous redérivons les résultats principaux du modèle bi-échelles du chapitre 4 pour le modèle multi-échelles présenté dans ce chapitre.

5.2.1 Implication du décentrage pour le modèle de BRDF

Dans la section 4.2.3, nous avons vu que toutes les équations de la théorie des micro-facettes découlent de la conservation des aires projetées : l'aire projetée de la surface géométrique dans la direction d'observation égale l'aire projetée de la micro-surface visible dans cette même direction, contrainte donnée par l'équation (4.4). Mais cette propriété n'est valide que lorsque la surface géométrique correspond exactement au plan moyen de la micro-surface, hypothèse implicite du modèle bi-échelles. Dans le modèle multi-échelles, ce n'est plus le cas, car le plan moyen de la micro-surface n'est pas la surface géométrique, mais la méso-surface. Les équations dérivées dans le chapitre 4 ne s'appliquent plus.

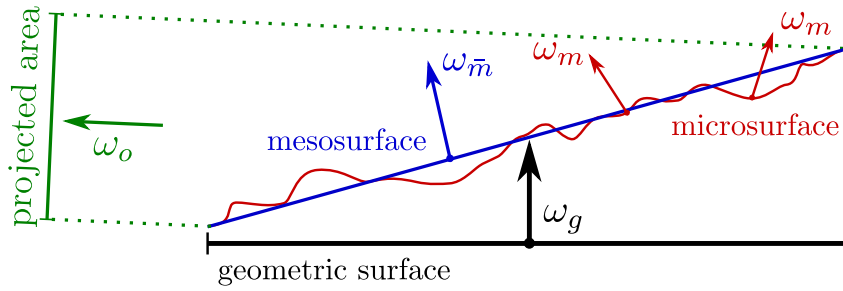


FIGURE 5.4 – Les équations de la théorie des micro-facettes sont dérivées en partant de la conservation de l'aire projetée. Dans la théorie multi-échelles, l'aire projetée de la micro-surface visible est équivalente à l'aire projetée de la méso-surface, et non à celle de la surface géométrique.

5.2.2 Conservation de l'aire projetée de la méso-surface

Pour le modèle multi-échelles, dans l'équation (4.4), le terme $\cos \theta_o$ correspondant à l'aire projetée de la surface géométrique, est remplacé par l'aire projetée de la méso-surface, comme illustré par la figure 5.4. L'aire projetée de la méso-surface dépend de sa normale, que nous appelons la *méso-normale* :

$$\omega_{\bar{m}} = \frac{(-\bar{x}_{\bar{m}}, -\bar{y}_{\bar{m}}, 1)}{\sqrt{\bar{x}_{\bar{m}}^2 + \bar{y}_{\bar{m}}^2 + 1}}.$$

et de son aire : $\frac{1}{\omega_{\bar{m}} \cdot \omega_g}$. L'aire projetée de la méso-surface est donnée par :

$$\text{projected area} = \frac{\omega_o \cdot \omega_{\bar{m}}}{\omega_{\bar{m}} \cdot \omega_g}.$$

L'équation (4.4) se réécrit :

$$\frac{\omega_o \cdot \omega_{\bar{m}}}{\omega_{\bar{m}} \cdot \omega_g} = \int_{\Omega} G_1(\omega_o, \omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m. \quad (5.1)$$

5.2.3 La fonction de masquage d'une micro-surface décentrée

Pour dériver la fonction de masquage sur une micro-surface décentrée, nous pourrions suivre la logique du chapitre 4, c'est-à-dire partir de l'équation 5.1 de la conservation de l'aire projetée et refaire toutes les dérivations jusqu'à obtenir la fonction de masquage. Mais ces dérivations sont longues et fastidieuses, en particulier celles de l'annexe A.1 qui donne l'expression de la fonction de masquage. Nous montrons qu'en exploitant une propriété d'invariance topologique de la fonction de masquage nous pouvons dériver le résultat directement de manière intuitive, sans passer par de telles dérivations.

Invariance de torsion de la fonction de masquage. Dans la section 4.5 nous avons vu que la fonction de masquage est invariante par étirement de la configuration. Similairement, la fonction de masquage est aussi invariante par torsion ("shear" en anglais) de la configuration, comme illustré en figure 5.5. Appliquer une torsion verticale sur la configuration équivaut à ajouter une valeur constante à toutes les pentes de la configuration. Comme pour l'étirement, cela inclut les pentes de la micro-surface, mais aussi la pente associée à la direction d'observation. Autrement dit, la fonction de masquage dans une configuration décentrée peut être exprimée comme la fonction de masquage dans une configuration centrée, en changeant simplement la pente de la direction d'observation.

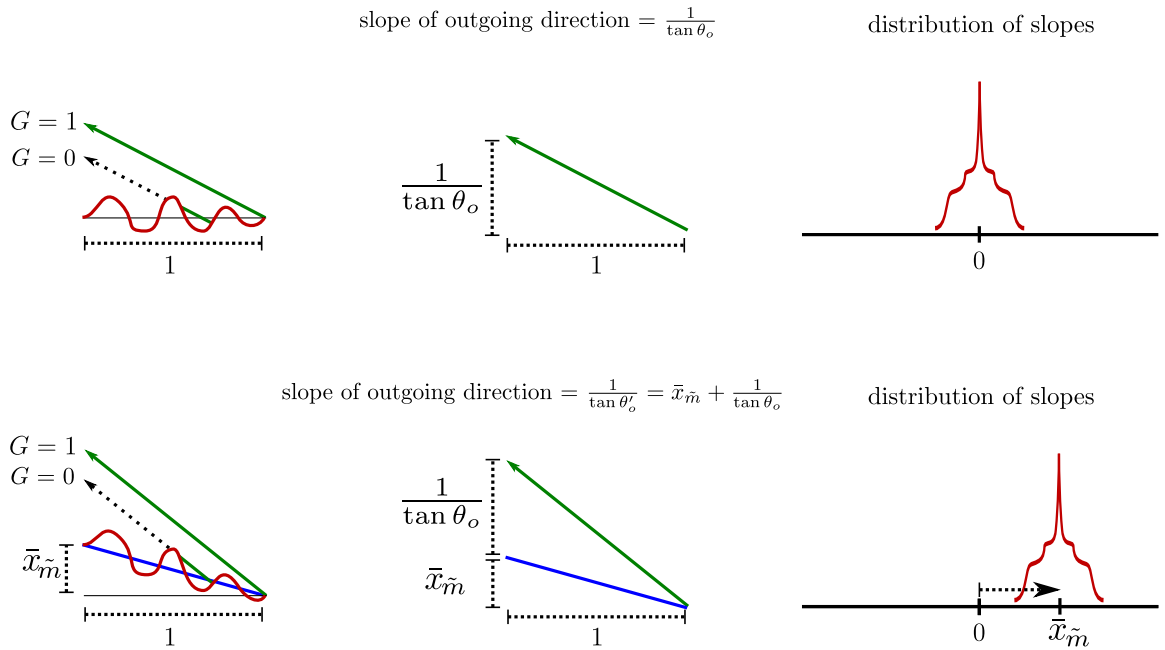


FIGURE 5.5 – Torsion (shear en anglais) verticale d'une configuration 1D. La torsion ne change pas la probabilité de masquage G_1 , mais à toutes les pentes de la configuration est additionné un facteur constant $\bar{x}_{\tilde{m}}$. Cela inclut les pentes de la micro-surface mais aussi la pente associée à la direction d'observation. La distribution de pentes est décalée d'un offset $\bar{x}_{\tilde{m}}$ et n'est plus centrée autour de 0.

Dérivation de la fonction de masquage décentrée. Nous avons vu en section 4.4.1 que la fonction de masquage de Smith s'exprime sous la forme :

$$\begin{aligned} G_1(\omega_o, \omega_m) &= G_1^{\text{local}}(\omega_o, \omega_m) G_1^{\text{dist}}(\omega_o), \\ G_1^{\text{local}}(\omega_o, \omega_m) &= \chi^+(\omega_o \cdot \omega_m), \\ G_1^{\text{dist}}(\omega_o) &= \frac{1}{1 + \Lambda\left(\frac{\|\mu_o\|}{\alpha_o}\right)}, \end{aligned}$$

où la pente de la direction d'observation est $\mu_o = \frac{(\cos \phi_o, \sin \phi_o)}{\tan \theta_o}$, telle que $\|\mu_o\| = \frac{1}{\tan \theta_o}$ et α_o est la rugosité projetée dans la direction d'observation, définie pour les distributions anisotropes en section 4.5. La fonction de masquage locale $G_1^{\text{local}}(\omega_o, \omega_m)$ peut être définie indépendamment des propriétés de la configuration puisqu'elle représente simplement le fait que la normale ω_m n'est pas auto-ombrée depuis ω_o . Nous pouvons donc laisser $G_1^{\text{local}}(\omega_o, \omega_m)$ sous cette forme, même avec des distributions décentrées. En revanche, la fonction distante $G_1^{\text{dist}}(\omega_o)$ dépend de la distribution des pentes et donc du décentrage. Mais cette probabilité est invariante par torsion. Donc, nous pouvons simplement recentrer la distribution des pentes en appliquant une torsion égale à la pente de la méso-surface, pour l'annuler et se ramener à une configuration centrée pour laquelle la fonction de masquage est connue. La pente d'observation après la torsion qui permet de recentrer la configuration est :

$$\mu'_o = \frac{(\cos \phi_o, \sin \phi_o)}{\tan \theta_o} - (\bar{x}_{\bar{m}}, \bar{y}_{\bar{m}}),$$

et la fonction de masquage distante dans le cas décentré est donnée par :

$$G_1^{\text{dist}}(\omega_o) = \frac{1}{1 + \Lambda\left(\frac{\|\mu'_o\|}{\alpha_o}\right)}.$$

Au final, la fonction de masquage pour les distributions décentrées vaut simplement :

$$G_1(\omega_o, \omega_m) = \frac{\chi^+(\omega_o \cdot \omega_m)}{1 + \Lambda\left(\frac{\|\mu'_o\|}{\alpha_o}\right)},$$

où, par rapport à l'équation (4.20), la fonction de masquage locale $G_1^{\text{local}}(\omega_o, \omega_m)$ n'a pas changé et on a juste translaté le vecteur de pente $\|\mu_o\|$ associé à ω_o dans la fonction de masquage distante $G_1^{\text{dist}}(\omega_o)$. Utiliser la propriété d'invariance par torsion nous a menés directement à un résultat simple, exact, et surtout, nous a économisé une dérivation "brute force à la main" des plus laborieuses !

5.2.4 La BRDF d'une micro-surface décentrée

Dans le chapitre précédent, toute la difficulté pour dériver correctement les équations des BRDFs à micro-facettes était de garantir la normalisation correcte, qui passe par la définition de la fonction de masquage. Maintenant que nous avons défini la fonction de masquage pour les distributions décentrées et que nous savons que le terme $\cos \theta_o = (\omega_o \cdot \omega_g)$ de l'aire projetée de la surface géométrique doit être remplacé par l'aire $\frac{\omega_o \cdot \omega_{\bar{m}}}{\omega_{\bar{m}} \cdot \omega_g}$ de la méso-surface, la modification des autres équations est plutôt facile à faire.

Les BRDFs à micro-facettes. L'équation (4.10) donne l'expression de la macro-BRDF du modèle bi-échelles. En remplaçant la fonction de masquage, les facteurs $|\omega_g \cdot \omega_o|$ et $|\omega_g \cdot \omega_i|$ par $|\omega_{\bar{m}} \cdot \omega_o|$ et $|\omega_{\bar{m}} \cdot \omega_i|$, et en divisant par l'aire de la méso-facette $\frac{1}{|\omega_g \cdot \omega_{\bar{m}}|}$, nous obtenons la méso-BRDF du modèle multi-échelles :

$$\rho(\omega_o, \omega_i) = \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o| |\omega_{\bar{m}} \cdot \omega_i|} \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m. \quad (5.2)$$

Avec cette définition de la méso-BRDF la radiance sortante s'évalue :

$$L(\omega_o) = \int_{\Omega} L(\omega_i) \rho(\omega_o, \omega_i) \langle \omega_i, \omega_{\bar{m}} \rangle d\omega_i,$$

où le cosinus de l'angle d'incidence est calculé par rapport à l'orientation de la méso-normale dans le terme $\langle \omega_i, \omega_{\bar{m}} \rangle$. À noter que nous aurions aussi pu choisir de laisser $\langle \omega_i, \omega_g \rangle$ au dénominateur de la BRDF et de la multiplier ensuite par ce terme plutôt que par $\langle \omega_i, \omega_{\bar{m}} \rangle$ dans l'intégrale avec l'éclairage. Ce choix est arbitraire, car dans les deux cas le terme se simplifie au moment de l'évaluation de la BRDF. Nous avons fait le choix d'écrire et d'intégrer la BRDF décentrée avec $\langle \omega_i, \omega_{\bar{m}} \rangle$ plutôt que $\langle \omega_i, \omega_g \rangle$ pour obtenir une expression réciproque dans l'équation (5.2).

La BRDF à micro-facettes spéculaires. L'équation (4.12) donne l'expression de la BRDF du modèle bi-échelles pour des micro-facettes spéculaires. En remplaçant la fonction de masquage, les facteurs $|\omega_g \cdot \omega_o|$ et $|\omega_g \cdot \omega_i|$ par $|\omega_{\bar{m}} \cdot \omega_o|$ et $|\omega_{\bar{m}} \cdot \omega_i|$, et en divisant par l'aire de la méso-facette $\frac{1}{|\omega_g \cdot \omega_{\bar{m}}|}$, nous obtenons la méso-BRDF du modèle multi-échelles :

$$\rho(\omega_o, \omega_i) = \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o| |\omega_{\bar{m}} \cdot \omega_i|} \frac{F(\omega_o, \omega_h) G_2(\omega_o, \omega_i, \omega_h) D(\omega_h)}{4}. \quad (5.3)$$

La BRDF à micro-facettes diffuses. L'équation (4.14) donne l'expression de la BRDF du modèle bi-échelles pour des micro-facettes diffuses. En remplaçant la fonction de masquage, les facteurs $|\omega_g \cdot \omega_o|$ et $|\omega_g \cdot \omega_i|$ par $|\omega_{\bar{m}} \cdot \omega_o|$ et $|\omega_{\bar{m}} \cdot \omega_i|$, et en divisant par l'aire de la méso-facette $\frac{1}{|\omega_g \cdot \omega_{\bar{m}}|}$, nous obtenons la méso-BRDF du modèle multi-échelles :

$$\rho(\omega_o, \omega_i) = \frac{1}{\pi} \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o| |\omega_{\bar{m}} \cdot \omega_i|} \int_{\Omega} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m. \quad (5.4)$$

5.2.5 Validation du modèle décentré

Nous souhaitons avoir un moyen de valider ces équations, pour être sûr de ne pas avoir fait d'erreur. Dans la section 4.3.5 nous avons proposé le *Weak White Furnace Test* comme critère de vérification pour valider les BRDFs à micro-facettes. Pour nous convaincre que les équations du modèle multi-échelles sont correctes, nous appliquons le même test. Dans le cas décentré, l'équation (4.15), en remplaçant le cosinus de l'angle d'incidence $\langle \omega_g, \omega_i \rangle$ par celui de l'angle par rapport à la méso-normal $\langle \omega_{\bar{m}}, \omega_i \rangle$, se réécrit :

$$\frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o|} \int_{\Omega} \frac{G_1(\omega_o, \omega_h) D(\omega_h)}{4} d\omega_i = 1. \quad (5.5)$$

et l'équation (4.16) :

$$\frac{1}{\pi} \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o|} \int_{\Omega} \int_{\Omega} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_1(\omega_o, \omega_m) D(\omega_m) d\omega_m d\omega_i = 1. \quad (5.6)$$

Ces équations sont satisfaites pour n'importe quelle distribution décentrée D et n'importe quelle direction d'observation ω_o . On peut s'en convaincre en réalisant une intégration numérique, par exemple.

5.3 Matériaux à nano-facettes

Nous souhaitons mettre au point des modèles multi-échelles où la micro-BRDF ρ_{μ} du matériau est non triviale. En effet, les équations (5.3) et (5.4) viennent de l'équation (5.2) générale du modèle à micro-facettes instanciée pour les cas particuliers où le matériau de la micro-surface est parfaitement spéculaire ou diffus. Dans cette section, nous instancions cette équation lorsque la micro-BRDF est elle-même une BRDF à micro-facettes (disons des *nano-facettes*) faites d'un matériau constant donné par la *nano-BRDF* $\rho_n(\omega_o, \omega_i, \omega_n)$.

Ceci n'était pas possible au chapitre précédent, car la distribution de normales nanoscopiques $D_{\omega_m}(\omega_n)$ d'une micro-facette est une distribution décentrée. En effet, si une micro-facette ω_m est faite d'une nano-surface, alors le plan moyen de cette nano-surface est celui de la micro-facette qui n'est pas nécessairement centré $\omega_m \neq \omega_g$. De fait, la micro-BRDF émergeant de la nano-surface est une BRDF à distribution décentrée. Nous avons donc besoin du modèle décentré présenté dans ce chapitre pour dériver une BRDF à micro-facettes à nano-facettes.

5.3.1 Le modèle de BRDF à nano-facettes

La micro-BRDF est une BRDF à micro-facettes décentrées donnée par l'équation 5.2, où nous remplaçons la méso-normale $\omega_{\bar{m}}$ par la normale de la micro-facette ω_m . Nous obtenons :

$$\rho_{\mu}(\omega_o, \omega_i, \omega_m) = \frac{|\omega_g \cdot \omega_m|}{|\omega_m \cdot \omega_o| |\omega_m \cdot \omega_i|} \int_{\Omega} \rho_n(\omega_o, \omega_i, \omega_n) \langle \omega_o, \omega_n \rangle \langle \omega_i, \omega_n \rangle G_{2\omega_m}(\omega_o, \omega_i, \omega_n) D_{\omega_m}(\omega_n) d\omega_n \quad (5.7)$$

Pour obtenir la macro-BRDF, nous développons l'expression de la micro-BRDF ρ_{μ} dans celle de l'expression de la méso-BRDF ρ de l'équation (5.2) :

$$\begin{aligned} \rho(\omega_o, \omega_i) &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o| |\omega_{\bar{m}} \cdot \omega_i|} \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_{2\omega_m}(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m \\ &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o| |\omega_{\bar{m}} \cdot \omega_i|} \\ &\quad \int_{\Omega} \frac{|\omega_g \cdot \omega_m|}{|\omega_m \cdot \omega_o| |\omega_m \cdot \omega_i|} \int_{\Omega} \rho_n(\omega_o, \omega_i, \omega_n) \langle \omega_o, \omega_n \rangle \langle \omega_i, \omega_n \rangle G_{2\omega_m}(\omega_o, \omega_i, \omega_n) D_{\omega_m}(\omega_n) d\omega_n \\ &\quad \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m. \end{aligned}$$

Nous pouvons simplifier les termes $\frac{1}{|\omega_m \cdot \omega_o| |\omega_m \cdot \omega_i|}$ et $\langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle$, car l'auto-ombrage représenté dans les produits scalaires clampés est toujours représenté dans la fonction de masquage et d'ombrage $G_2(\omega_o, \omega_i, \omega_m)$ et nous obtenons :

$$\begin{aligned} \rho(\omega_o, \omega_i) &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o| |\omega_{\bar{m}} \cdot \omega_i|} \\ &\quad \int_{\Omega} |\omega_g \cdot \omega_m| \int_{\Omega} \rho_n(\omega_o, \omega_i, \omega_n) \langle \omega_o, \omega_n \rangle \langle \omega_i, \omega_n \rangle G_2(\omega_o, \omega_i, \omega_m) G_{2\omega_m}(\omega_o, \omega_i, \omega_n) D_{\omega_m}(\omega_n) D(\omega_m) d\omega_n d\omega_m. \end{aligned} \quad (5.8)$$

Distribution de normales de la nano-surface totale. On note $D_{f\omega_n}(\omega_n)$ la distribution de nano-normales de l'ensemble des micro-facettes :

$$D_{f\omega_n}(\omega_n) = \int_{\Omega} |\omega_g \cdot \omega_m| D_{\omega_m}(\omega_n) D(\omega_m) d\omega_m. \quad (5.9)$$

Cette définition est intuitivement similaire à une convolution, mais n'en est pas une au sens strict (la convolution n'est pas définie dans le cas général sur la sphère).

Masquage et ombrage de la nano-surface totale. Dans l'équation précédente, la fonction $G_2(\omega_o, \omega_i, \omega_m)$ est la probabilité de masquage et d'ombrage d'une micro-facette par l'ensemble de la micro-surface, et la fonction $G_{2\omega_m}(\omega_o, \omega_i, \omega_n)$ est la probabilité d'ombrage et de masquage d'une nano-facette par la nano-surface *de la même micro-facette*. La probabilité d'ombrage et de masquage d'une nano-facette par la nano-surface *totale*, c'est-à-dire la probabilité que cette nano-facette soit visible en même temps depuis ω_o et ω_i depuis l'extérieur de la surface, est donnée par :

$$G_{2f\omega_n}(\omega_o, \omega_i, \omega_n) = \frac{\int_{\Omega} G_2(\omega_o, \omega_i, \omega_m) G_{2\omega_m}(\omega_o, \omega_i, \omega_n) D_{\omega_m}(\omega_n) D(\omega_m) d\omega_m}{\int_{\Omega} D_{\omega_m}(\omega_n) D(\omega_m) d\omega_m}.$$

Le modèle de BRDF à nano-facettes. En utilisant les définitions des deux équations précédentes, nous réécrivons l'équation (5.8) :

$$\rho(\omega_o, \omega_i) = \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_{\bar{m}} \cdot \omega_o| |\omega_{\bar{m}} \cdot \omega_i|} \int_{\Omega} \rho_n(\omega_o, \omega_i, \omega_n) \langle \omega_o, \omega_n \rangle \langle \omega_i, \omega_n \rangle G_{2 \int \omega_n}(\omega_o, \omega_i, \omega_m) D_{f \omega_n}(\omega_n) d\omega_n, \quad (5.10)$$

qui n'est autre que l'équation (5.2) des BRDFs à micro-facettes décentrées, où on utilise la distribution de normales et la fonction d'ombrage et de masquage de la nano-surface totale.

Enfin, puisque la distribution de nano-normales D_{ω_m} est centrée sur la micro-facette ω_m , la distribution totale des nano-normales $D_{f \omega_n}$ est centrée sur la micro-surface moyenne donnée par $\omega_{\bar{m}}$. De fait, dans le cas particulier où la distribution de micro-normales est centrée $\omega_{\bar{m}} = \omega_g$, la distribution de normales de la nano-surface totale est centrée aussi et cette équation devient :

$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o| |\omega_g \cdot \omega_i|} \int_{\Omega} \rho_n(\omega_o, \omega_i, \omega_n) \langle \omega_o, \omega_n \rangle \langle \omega_i, \omega_n \rangle G_{2 \int \omega_n}(\omega_o, \omega_i, \omega_m) D_{f \omega_n}(\omega_n) d\omega_n, \quad (5.11)$$

et on retrouve l'équation (4.10) classique des modèles à micro-facettes.

Conclusion. Le modèle de BRDF à micro-facettes est un modèle clos : une BRDF à micro-facettes dont le matériau des micro-facettes est aussi une BRDF à micro-facettes peut s'exprimer directement sous la forme classique d'une BRDF à micro-facettes. Cette propriété est très importante en pratique, car elle permet de modéliser des matériaux de micro-surfaces non triviaux tout en se ramenant à l'expression standard du modèle. Par exemple, lorsque la micro-BRDF est un matériau à micro-facettes spéculaires (ou diffuses), alors la macro-BRDF est aussi une BRDF à micro-facettes spéculaires (ou diffuses).

La partie difficile pour utiliser ce modèle est de déterminer les expressions de $D_{f \omega_n}(\omega_n)$ et de $G_{2 \int \omega_n}(\omega_o, \omega_i, \omega_n)$. C'est l'objet de la section suivante pour le cas particulier des height maps.

5.3.2 Le modèle de BRDF à nano-facettes pour les height maps

La définition de la distribution de normales de l'équation (5.9) et de la fonction de masquage et d'ombrage de l'équation (5.10), sont difficiles à évaluer directement, car elles n'ont pas de formes closes dans le cas général. Nous allons voir que ces deux fonctions ont une forme close que nous utiliserons en pratique dans les parties applications, dans le cas particulier où :

1. Les distributions de normales sont des distributions de pentes, c'est-à-dire que la micro-surface et la nano-surface sont toutes les deux une height map (le cas qui nous intéresse dans le cadre de cette thèse).
2. La distribution des nano-facettes est identique au décentrage près, sur chaque micro-facette ω_m , c'est-à-dire qu'elle s'écrit sous la forme $P_{\omega_m}^{22} = P_{\mu}^{22}(x_{\bar{n}} - x_{\bar{m}}, y_{\bar{n}} - y_{\bar{m}})$, où P_{μ}^{22} est une distribution constante et $(x_{\bar{m}}, y_{\bar{m}})$ sont les pentes de la micro-facette.
3. Le profil de la nano-surface totale est connu (par exemple, c'est celui de Smith ou des V-cavités).

Distributions de pentes. Les distributions des pentes de la micro-surface, de la nano-surface d'une micro-facette et de la nano-surface totale sont telles que :

$$\begin{aligned} D(\omega_m) &= \frac{P^{22}(x_{\bar{n}}, y_{\bar{m}})}{|\omega_g \cdot \omega_m|^4}, \\ D_{\omega_m}(\omega_n) &= \frac{P_{\omega_n}^{22}(x_{\bar{n}}, y_{\bar{n}})}{|\omega_g \cdot \omega_n|^4} = \frac{P_{\mu}^{22}(x_{\bar{n}} - x_{\bar{m}}, y_{\bar{n}} - y_{\bar{m}})}{|\omega_g \cdot \omega_n|^4}, \\ D_{f \omega_n}(\omega_n) &= \frac{P_{f \omega_n}^{22}(x_{\bar{n}}, y_{\bar{n}})}{|\omega_g \cdot \omega_n|^4}. \end{aligned}$$

L'équation (5.9) devient :

$$\frac{P_{f\omega_n}^{22}(x_{\bar{n}}, y_{\bar{n}})}{|\omega_g \cdot \omega_n|^4} = \int_{\Omega} |\omega_g \cdot \omega_m| \frac{P_{\mu}^{22}(x_{\bar{n}} - x_{\bar{m}}, y_{\bar{n}} - y_{\bar{m}})}{|\omega_g \cdot \omega_n|^4} \frac{P^{22}(x_{\bar{m}}, y_{\bar{m}})}{|\omega_g \cdot \omega_m|^4} d\omega_m$$

ce qui en simplifiant $\frac{1}{|\omega_g \cdot \omega_n|^4}$ et $|\omega_g \cdot \omega_m|$ devient :

$$P_{f\omega_n}^{22}(x_{\bar{n}}, y_{\bar{n}}) = \int_{\Omega} P_{\mu}^{22}(x_{\bar{n}} - x_{\bar{m}}, y_{\bar{n}} - y_{\bar{m}}) \frac{P^{22}(x_{\bar{m}}, y_{\bar{m}})}{|\omega_g \cdot \omega_m|^3} d\omega_m$$

en changeant l'espace d'integration, avec le jacobien $d\omega_m = |\omega_g \cdot \omega_m|^3 dx_{\bar{m}} dy_{\bar{m}}$, on obtient :

$$\begin{aligned} P_{f\omega_n}^{22}(x_{\bar{n}}, y_{\bar{n}}) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P_{\mu}^{22}(x_{\bar{n}} - x_{\bar{m}}, y_{\bar{n}} - y_{\bar{m}}) P^{22}(x_{\bar{m}}, y_{\bar{m}}) dx_{\bar{m}} dy_{\bar{m}} \\ &= (P_{\mu}^{22} * P^{22})(x_{\bar{n}}, y_{\bar{n}}), \end{aligned} \quad (5.12)$$

le produit de convolution de la distribution des pentes des micro-facettes et de la distribution des pentes des nano-facettes.

Fonction de masquage et d'ombrage de la nano-surface totale. Nous connaissons désormais la distribution des pentes $P_{f\omega_n}^{22}$ de la nano-surface totale. Sous l'hypothèse 3 que la nano-surface totale a un profil de surface connu, alors sa fonction de masquage et d'ombrage est entièrement déterminée, comme nous l'avons vu dans le chapitre 4 et se calcule en fonction de $P_{f\omega_n}^{22}$. Dans les chapitres 8 et 9 nous proposons une manière de représenter P^{22} et P_{μ}^{22} qui permet d'évaluer facilement la convolution $P_{f\omega_n}^{22} = (P_{\mu}^{22} * P^{22})$ en pratique.

5.4 Résumé du chapitre

Points clés du chapitre

- La théorie standard des micro-facettes est une théorie bi-échelles ne faisant intervenir que l'échelle macroscopique (la surface géométrique) et l'échelle microscopique (la micro-surface).
- La mise au point d'un modèle multi-échelles nécessite l'introduction d'une échelle de transition intermédiaire : l'échelle mésoscopique.
- La surface mésoscopique correspond à la pente moyenne de la micro-surface, qui est différente de la macro-surface dans le cas général. Nous qualifions de "décentrée" ce type de micro-surface.
- Certaines approches précédentes consistent à redéfinir, par un changement de repère, la surface macroscopique comme étant la surface mésoscopique, pour se retrouver dans une configuration centrée où la BRDF est définie et évaluable. Cela est mal posé et produit un résultat faux.
- Pour calculer un résultat correct, nous proposons de dériver les équations du modèle décentré.
- Dans le modèle multi-échelles, l'aire projetée de la micro-surface visible est égale à l'aire projetée de la méso-surface et non à celle de la macro-surface. Nous partons de ce constat pour dériver les équations du modèle multi-échelles de la même manière que pour le modèle bi-échelles.
- En remarquant et en utilisant l'invariance par torsion de la fonction de masquage nous dérivons intuitivement les équations du modèle multi-échelles sans passer par des dérivations mathématiques complexes.
- Nous validons les équations du modèle multi-échelles en utilisant le *Weak White Furnace Test* que nous avons proposé dans le chapitre 4.

Chapitre 6

Etude de la corrélation azimuthale entre le masquage et l'ombrage dans le modèle de Smith

Dans ce chapitre, nous proposons un modèle pour la fonction de masquage et d'ombrage $G_2(\omega_o, \omega_i, \omega_m)$ qui tient compte de la corrélation azimuthale entre le masquage $G_1(\omega_o, \omega_m)$ et l'ombrage $G_1(\omega_i, \omega_m)$. La figure 6.1 illustre cet effet : lorsque deux directions incidentes sont proches, les parties de la surface visibles depuis une direction tendent à être les mêmes que celles qui sont visibles depuis l'autre direction. Cette effet porte parfois le nom de *hot spot* : lorsque la direction d'observation et la direction de l'éclairage se rapprochent, les ombres semblent disparaître pour l'observateur car elles ne sont plus dans des zones visibles pour lui : ce qui est visible pour la lumière, donc non ombré, l'est aussi pour l'observateur.

Nous avons vu en section 3.1.3 qu'il existe plusieurs modèles de la fonction de masquage et d'ombrage $G_2(\omega_o, \omega_i, \omega_m)$. Mais il n'existe pas de modèle physique dédié à modéliser cette corrélation azimuthale, les modèles existants étant purement empiriques. Dans ce chapitre, nous proposons un modèle pour cette corrélation et en dérivons une approximation analytique précise et facile à intégrer dans le modèle existant de la fonction de masquage et d'ombrage de Smith.

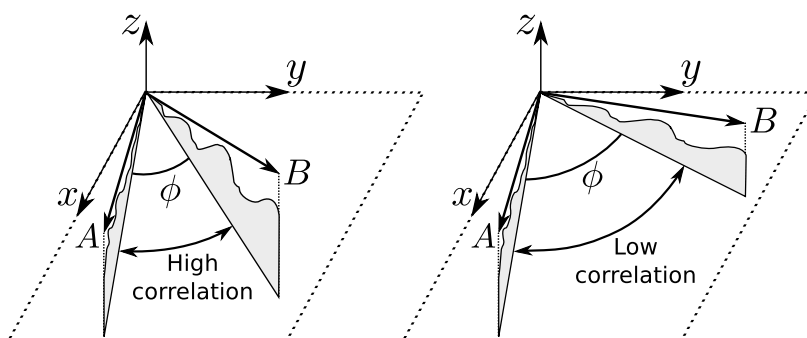


FIGURE 6.1 – Illustration de la corrélation azimuthale entre les fonctions de masquage de deux directions A et B : plus les directions sont proches azimuthalement, plus la probabilité que la partie de la surface visible pour A le soit aussi pour B est importante, et réciproquement.

6.1 Introduction

Idée principale du chapitre. Par souci de cohérence, nous avons cherché à écrire ce chapitre avec des notations au maximum compatibles avec celles des autres chapitres. Mais celui-ci est particulier dans le sens où, contrairement aux autres, il est massivement basé sur des dérivations de probabilités conditionnelles qui exigent des notations spécifiques et parfois assez lourdes.

Bien que les notations et les dérivations de ce chapitre semblent compliquées, le résultat auquel nous arrivons est très simple, à la fois à comprendre et à utiliser en pratique. En résumé, le nouveau modèle de fonction de masquage et d’ombrage que nous proposons est donné sous une forme similaire à celle de l’équation (3.8) du chapitre 3 :

$$G_2(\omega_o, \omega_i, \omega_m) = \frac{\chi^+(\omega_o \cdot \omega_m) \chi^+(\omega_i \cdot \omega_m)}{1 + \max(\Lambda(\omega_o), \Lambda(\omega_i)) + r_0(\omega_o, \omega_i) \min(\Lambda(\omega_o), \Lambda(\omega_i))},$$

où nos contributions résident en la mise au point d’un modèle pour le facteur de corrélation $r_0(\omega_o, \omega_i)$ et en une approximation simple pour l’évaluer en pratique.

Organisation du chapitre. Notre approche est basée sur celle du papier original de Smith [Smi67]. Nous supposons une micro-surface 2D isotrope dont les hauteurs et les pentes sont des variables aléatoires indépendantes à distributions gaussiennes. Pour introduire les notations de probabilité utilisées dans ce chapitre, nous rappelons le calcul de la fonction de masquage de Smith [Smi67] et son extension bistatique proposée par Bourlier et al. [BBS02, BB03] en section 6.2. En sections 6.3 et 6.4, nous décrivons et développons notre modèle bistatique qui incorpore la corrélation azimuthale entre le masquage et l’ombrage sous la forme d’un facteur de corrélation r_0 . Dans la section 6.4.5, nous proposons une approximation analytique pour évaluer r_0 .

Compatibilité du modèle avec le contenu des autres chapitres. Dans les chapitres de la partie II, nous construisons une application de rendu de surfaces multi-échelles dans laquelle nous utilisons essentiellement des distributions de Beckmann anisotropes et décentrées pour représenter la surface à travers les échelles. Il est donc important que le modèle présenté dans ce chapitre soit compatible avec une telle micro-surface. Bien que les dérivations que nous allons faire dans ce chapitre supposent une micro-surface à statistiques de pentes gaussiennes isotropes et centrées, ce ne sont pas des limitations.

En effet, nous avons vu dans la section 4.5 que la fonction de masquage est invariante par étirement, ce qui permet de généraliser les résultats obtenus pour des micro-surfaces à statistiques isotropes à des micro-surfaces à statistiques de la même classe mais anisotropes. Il en va de même pour la fonction de masquage et d’ombrage : si elle est connue pour des distributions de Beckmann isotropes, alors la propriété d’invariance par étirement peut être utilisée pour généraliser ce résultat à des distributions de Beckmann anisotropes.

De la même manière, dans la section 5.2, nous avons montré que les résultats obtenus pour les micro-surfaces centrées peuvent se généraliser aux micro-surfaces décentrées en utilisant l’invariance par torsion de la fonction de masquage. Il en va de même avec la fonction de masquage et d’ombrage : si elle est connue pour une micro-surface centrée, alors elle est connue pour une micro-surface décentrée.

En conclusion, il est tout à fait possible d’utiliser le résultat introduit dans ce chapitre avec des distributions anisotropes et décentrées, le rendant compatible avec l’ensemble des chapitres de la partie II.

6.2 Description de la configuration

6.2.1 Statistiques de la micro-surface

Un point $M(h, \tilde{m})$ de la micro-surface est décrit par sa hauteur h et sa pente $\tilde{m} = (x_{\tilde{m}}, y_{\tilde{m}})$. Dans le modèle de micro-surface de Smith, les hauteurs et les pentes sont modélisées comme des variables aléatoires indépendantes qui suivent des lois normales :

$$h \sim \mathcal{N}(0, \sigma_h^2), \tilde{m} \sim \mathcal{N}((0, 0), \sigma_{\tilde{m}}^2),$$

où $\mathcal{N}(\mu, \sigma^2)$ est la loi normale de moyenne μ et de variance σ^2 . Les densités de probabilité associées sont :

$$P^1(h) = \frac{1}{\sqrt{2\pi} \sigma_h} \exp\left(-\frac{h^2}{2\sigma_h^2}\right), P^{22}(\tilde{m}) = \frac{1}{2\pi \sigma_{\tilde{m}}^2} \exp\left(-\frac{x_{\tilde{m}}^2 + y_{\tilde{m}}^2}{2\sigma_{\tilde{m}}^2}\right).$$

On note $C^1(h) = \int_{-\infty}^h P^1(h') dh'$, la fonction de répartition (CDF) de h . A noter que la distribution P^{22} des pentes est une distribution de Beckmann dont la rugosité vaut $\alpha = \sqrt{2}\sigma_{\tilde{m}}$.

6.2.2 Notations probabilistes

Comme notre dérivation est fortement basée sur la théorie des probabilités, nous adoptons dans ce chapitre une terminologie et des notations probabilistes. Les notations des événements et de leurs probabilités associées sont données en tables 6.1 et 6.2.

Probabilité	Événement associé
$G_1(A, M = (h, \tilde{m}))$	Le point M est visible depuis la direction A
$G_1^{\text{local}}(A, \tilde{m})$	Le point M n'est pas auto-masqué
$G_1^{\text{dist}}(A, h)$	Le rayon "MA" est au-dessus de la surface en $\tau_A \in]0, \infty[$
$G_1^{\text{dist}}(\bar{A}_{\tau_A} A_{\tau_A - d\tau}, h)$	Le rayon "MA" intersecte la surface en τ_A sachant qu'il est au-dessus de la surface en $\tau_A - d\tau$

TABLE 6.1 – Notations utilisées pour la fonction de masquage.

Probabilité	Événement associé
$G_2(A \cap B, M)$	Le point M est visible depuis les directions A et B
$G_2^{\text{local}}(A \cap B, \tilde{m})$	Le point M n'est pas auto-ombré pour les directions A et B
$G_2^{\text{dist}}(A \cap B, h)$	Le rayon "MA" est au-dessus de la surface en $\tau_A \in]0, \infty[$ et le rayon "MB" est au-dessus de la surface en $\tau_B \in]0, \infty[$
$G_1^{\text{dist}}(B A, h)$	Le rayon "MB" est au-dessus de la surface en $\tau_B \in]0, \infty[$ sachant que le rayon "MA" est au-dessus de la surface en $\tau_A \in]0, \infty[$
$G_1^{\text{dist}}(\bar{B}_{\tau_B} (B_{\tau_B - d\tau} \cap A), h)$	Le rayon "MB" intersecte la surface en τ_B sachant qu'il est au-dessus de la surface en $\tau_B - d\tau$ et que le rayon "MA" est au-dessus de la surface en $\tau_A \in]0, \infty[$

TABLE 6.2 – Notations utilisées pour la fonction de masquage et d'ombrage.

6.2.3 La fonction de masquage

La fonction de masquage $G_1(A, M)$ donne la probabilité qu'un point de la surface $M(h, \tilde{m})$, de hauteur h et de pente \tilde{m} , soit visible depuis une direction A , dont l'angle d'incidence est θ_A et dont la pente associée à la direction d'observation est donnée par $\mu_A = \cot \theta_A$. Comme nous l'avons vu en section 4.4.1, elle s'exprime comme le produit de deux probabilités :

$$G_1(A, M = (h, \tilde{m})) = G_1^{\text{dist}}(A, h) G_1^{\text{local}}(A, \omega_m),$$

où la fonction $G_1^{\text{local}}(A, \omega_m)$ donne la probabilité de masquage "locale". Elle représente le fait que le point ne soit pas masqué par lui-même, c'est-à-dire qu'il ne soit pas auto-masqué ($A \cdot \omega_m < 0$) :

$$\begin{aligned} G_1^{\text{local}}(A, \tilde{m}) &= G_1^{\text{local}}(A, \omega_m) \\ &= \chi^+(A \cdot \omega_m). \end{aligned} \quad (6.1)$$

La fonction $G_1^{\text{dist}}(A, h)$ donne la probabilité de masquage "distante", c'est-à-dire la probabilité d'occlusion par un autre point de la surface du rayon partant de M dans la direction A , et elle est donnée dans le modèle de Smith par :

$$G_1^{\text{dist}}(A, h) = \exp \left[- \int_0^\infty G_1^{\text{dist}}(\bar{A}_{\tau_A} | A_{\tau_A - d\tau}, h) d\tau_A \right], \quad (6.2)$$

où la probabilité que le rayon "MA" intersecte la surface en τ_A sachant qu'il ne l'intersecte pas en $\tau_A - d\tau$ est donnée par :

$$G_1^{\text{dist}}(\bar{A}_{\tau_A} | A_{\tau_A - d\tau}, h) = \Lambda(\mu_A) \frac{\mu_A P^1(h + \tau_A \mu_A)}{C^1(h + \tau_A \mu_A)}. \quad (6.3)$$

A noter que $G_1^{\text{dist}}(\bar{A}_{\tau_A} | A_{\tau_A - d\tau}, h)$ est souvent notée g dans la littérature. Nous avons changé cette notation par souci de cohérence avec le reste du chapitre. L'équation (6.2) a une solution analytique :

$$G_1^{\text{dist}}(A, h) = C^1(h)^{\Lambda(\mu_A)}. \quad (6.4)$$

En théorie des micro-facettes, nous nous intéressons souvent à la fonction de masquage exprimée comme une fonction des normales mais pas comme une fonction des hauteurs. Pour éliminer la variable h , nous calculons la fonction de masquage moyennée sur les hauteurs :

$$\begin{aligned} \bar{G}_1(A, \tilde{m}) &= G_1^{\text{local}}(A, \omega_m) \int_{-\infty}^{+\infty} G_1^{\text{dist}}(A, h) P^1(h) dh \\ &= G_1^{\text{local}}(A, \omega_m) \int_{-\infty}^{+\infty} C^1(h)^{\Lambda(\mu_A)} P^1(h) dh \\ &= \chi^+(A \cdot \omega_m) \frac{1}{1 + \Lambda(A)}, \end{aligned}$$

qui est la fonction de masquage de Smith sous sa forme utilisée dans les BRDFs à micro-facettes.

6.2.4 La fonction de masquage et d'ombrage

Dans une configuration bistatique, comme illustré dans la figure 6.1, nous considérons une seconde direction d'observation B , d'angle d'incidence θ_B et de pente $\mu_B = \cot \theta_B$. Comme nous considérons une surface isotrope, nous définissons la différence azimuthale : $\phi = \min(|\phi_A - \phi_B|, |\mu_A - \mu_B - 2\pi|)$ et $\phi \in [0, \pi]$.

La fonction de masquage bistatique (ou de "masquage et d'ombrage") donne la probabilité qu'un point de la surface soit visible depuis les directions A et B *en même temps*. On distingue deux cas extrêmes. Dans le premier cas $\phi = 0$, c'est-à-dire que A et B sont dans un même plan orthogonal à la surface. Dans ce cas,

les probabilités de masquage d'un point M depuis A et B sont corrélées, puisque si M est visible pour la direction de pente μ la plus basse, alors il est aussi visible pour l'autre direction. De fait, seule la plus petite probabilité compte :

$$G_2(A \cap B, M) = \min(G_1(A, M), G_1(B, M)). \quad (6.5)$$

Dans le second cas où $\phi \in]0, \pi]$, si nous supposons que les événements associés à A et à B sont indépendants, alors les probabilités de masquage peuvent être multipliées :

$$G_2(A \cap B, M) = G_1(A, M) G_1(B, M). \quad (6.6)$$

En général, lorsque A et B sont proches (figure 6.1), les événements de masquage peuvent être fortement corrélés et leur probabilités ne peuvent pas être multipliées. Cela se produit lorsque la différence azimuthale ϕ est petite. L'évaluation de la fonction d'ombrage et de masquage est difficile dans ce cas :

$$G_2(A \cap B, M) = ?$$

Idée. Comme les hauteurs et les pentes sont modélisées par des variables aléatoires indépendantes, nous pouvons commencer par écrire :

$$G_2(A \cap B, M) = G_2^{\text{dist}}(A \cap B, h) G_2^{\text{local}}(A \cap B, \tilde{m}) \quad (6.7)$$

La fonction de masquage et d'ombrage locale $G_2^{\text{local}}(A \cap B, \tilde{m})$ qui donne la probabilité qu'une facette de pente \tilde{m} ne soit pas auto-masquée simultanément depuis les directions A et B est facile à modéliser, nous l'expliquons dans la section 6.3. La modélisation difficile étudiée dans ce chapitre est celle qui concerne la fonction de masquage et d'ombrage distante $G_2^{\text{dist}}(A \cap B, h)$, elle est développée dans la section 6.4.

6.3 Dérivation de la fonction de masquage et d'ombrage locale

Un point de la surface de pente \tilde{m} n'est pas auto-masqué depuis les directions A et B seulement si sa normale a un produit scalaire positif avec les deux directions : $A \cdot \omega_m > 0$ et $B \cdot \omega_m > 0$. Nous écrivons cette condition avec deux fonctions de Heaviside :

$$G_2^{\text{local}}(A \cap B, \tilde{m}) = \chi^+(A \cdot \omega_m) \chi^+(B \cdot \omega_m).$$

La fonction de masquage locale moyennée pour l'ensemble des pentes de la surface est donnée par :

$$\bar{G}_2^{\text{local}}(A \cap B) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \chi^+(A \cdot \omega_m) \chi^+(B \cdot \omega_m) P^{22}(\tilde{m}) d\tilde{m}. \quad (6.8)$$

Cette fonction moyennée sur les normales n'a pas de solution analytique. En pratique, ce n'est pas un problème car elle n'est jamais utilisée dans les modèles de BRDFs, comme nous l'avons expliqué à la fin de la section 4.4.1 (paragraphe "Les fonctions de masquage de Smith moyennées"). Dans ce chapitre, nous l'intégrons numériquement pour les comparaisons du modèle développé dans la section 10.3 où nous présentons des résultats.

6.4 Dérivation de la fonction de masquage et d'ombrage distante

La probabilité que des rayons partant de M dans les directions A et B n'intersectent pas la surface peut être développée avec le théorème de Bayes par :

$$G_2^{\text{dist}}(A \cap B, h) = G_1^{\text{dist}}(A, h) G_1^{\text{dist}}(B | A, h), \quad (6.9)$$

où $G_1^{\text{dist}}(A, h)$ est la fonction de masquage monostatique de l'équation(6.2), et la probabilité conditionnelle $G_1^{\text{dist}}(B | A, h)$ est associée à l'événement suivant : le rayon "MB" n'intersecte pas la surface sachant que le rayon "MA" n'intersecte pas la surface non plus. Le but de cette section est de trouver une manière de modéliser cette probabilité conditionnelle.

L'événement A : le rayon "MA" n'intersecte pas la surface donne l'information suivante : "Tous les points de la surface sous A ont des hauteurs bornées par la hauteur de A à l'abscisse correspondante". Nous introduisons cet événement dans l'équation (6.2) :

$$G_1^{\text{dist}}(B | A, h) = \exp \left[- \int_0^\infty G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A), h) d\tau_B \right], \quad (6.10)$$

où nous avons remplacé la fonction $G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h)$ par $G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A), h)$ en utilisant l'information donnée par l'événement A .

Modéliser l'information donnée par l'événement A n'est pas simple pour deux raisons. D'une part, l'information n'est pas ponctuelle (un seul point de la surface) mais 1D (l'ensemble de points de la surface le long de A) et est donc difficile à modéliser et à utiliser. D'autre part, cette information est liée au comportement de la surface et doit être exprimée d'une manière qui a du sens pour le calcul de l'intersection entre la surface et le rayon "MB". Par la suite, nous proposons un modèle pour cette information et nous montrons comment l'utiliser dans le calcul de la probabilité $G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A), h)$.

6.4.1 Dérivation de la fonction de masquage distante conditionnelle

Pour chaque point du rayon "MB", plutôt que d'utiliser toute l'information disponible sous le rayon "MA", nous utilisons uniquement l'information donnée par le point de "MA" dans la direction orthogonale au rayon "MB" le long de la surface, comme illustré en figure 6.2.

On note τ_A l'abscisse de l'unique point de "MA" considéré et on approxime :

$$G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A), h) \approx G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A_{\tau_A}), h).$$

Une conséquence importante pour le modèle est que nous devons choisir A et B tels que $\mu_A \leq \mu_B$. En effet, dans une configuration planaire, si $\mu_A > \mu_B$ alors $B_{\tau_B-d\tau} \Rightarrow A_{\tau_A}$ et donc :

$$G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A_{\tau_A}), h) = G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h),$$

et l'information donnée par le rayon "MA" n'est pas utilisée. A partir de maintenant, nous allons toujours supposer que $\mu_A \leq \mu_B$. Cette hypothèse ne fait pas perdre le modèle en généralité, puisque la fonction de masquage bistatique est commutative.

A noter que cette approximation est très proche de celle utilisée dans le modèle de Smith [Smi67] pour la dérivation de la fonction de masquage monostatique. En effet, dans le cas monostatique, la probabilité $G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h)$ de l'équation (6.3) donne la probabilité que le rayon intersecte la surface à la distance τ_B sachant que le rayon est au-dessus de la surface à la distance $\tau_B - d\tau$. Choisir l'événement : la surface est en-dessous du rayon à l'abscisse $\tau_B - d\tau$, est une approximation de l'événement : la surface est en-dessous du rayon dans l'intervalle $[0; \tau_B[$. Comme modéliser l'information sur l'intervalle $[0; \tau_B[$ est difficile, Smith remplace cette information par celle donnée par un seul point situé à $\tau_B - d\tau$. Il a remplacé l'événement $B_{[0, \tau_B[}$ par $B_{\tau_B-d\tau}$. Notre approximation est similaire, mais est appliquée sur l'autre rayon. On choisit un point sous "MA" dans la direction orthogonale à "MB" de manière à avoir $A_{\tau_A} | B_{\tau_B-d\tau} \approx A_{\tau_A} | B_{\tau_B}$, ce qui nous permet de simplifier l'équation (6.11).

Si $\phi \in [\frac{\pi}{2}, \pi]$, alors pour les abscisses $\tau_B \geq 0$, le point sous "MA" est localisé à une distance infinie et n'apporte pas d'information utilisable. Dans ce cas, on néglige l'information donnée par l'événement A et on a $G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A_{\tau_A}), h) = G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h)$. Par la suite, on considère les cas où $\phi \in [0, \frac{\pi}{2}[$.

Si $\phi \in [0, \frac{\pi}{2}[$, le point sous "MA" correspondant à l'abscisse τ_B sous "MB" a une abscisse $\tau_A = \tau_B / \cos \phi$ sous A , et la distance entre les deux points est $d = \tau_B \tan \phi$ (voir la figure 6.3). Par ailleurs, comme le point

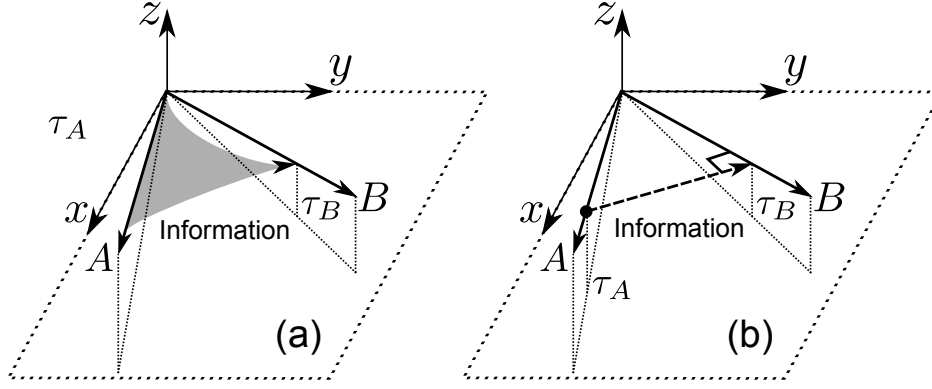


FIGURE 6.2 – (a) La visibilité depuis la direction A introduit une contrainte sur les hauteurs de tous les points de la surface sous le rayon “ MA ”. (b) Dans le calcul de la probabilité de masquage de B , nous approximations cette information en ne tenant compte que d’un seul point sous “ MA ”, celui qui correspond à la projection orthogonale sur le rayon “ MB ” au point d’abscisse τ_B .

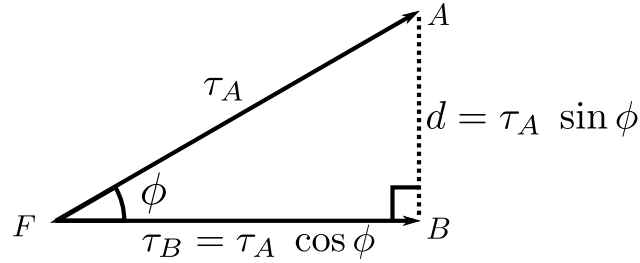


FIGURE 6.3 – Distance entre les points d’abscisse τ_A et τ_B respectivement sous les rayons “ MA ” et “ MB ”.

est en-dessous du rayon “ MA ”, sa hauteur est bornée par $h_{\tau_A} < h + \tau_A \mu_A = h + \tau_B \mu_A / \cos \phi$. On note A_{τ_A} cet événement. En utilisant le théorème de Bayes, nous développons :

$$\begin{aligned}
 G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A_{\tau_A}), h) &= G_1^{\text{dist}}((\bar{B}_{\tau_B} | A_{\tau_A}) | B_{\tau_B-d\tau}, h) \\
 &= G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h) \frac{G_1^{\text{dist}}((A_{\tau_A} | \bar{B}_{\tau_B}) | B_{\tau_B-d\tau}, h)}{G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B-d\tau}, h)} \\
 &= G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h) \frac{G_1^{\text{dist}}(A_{\tau_A} | (\bar{B}_{\tau_B} \cap B_{\tau_B-d\tau}), h)}{G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B-d\tau}, h)}.
 \end{aligned}$$

Pour simplifier cette équation, nous utilisons le fait que $\bar{B}_{\tau_B} \cap B_{\tau_B-d\tau} = \bar{B}_{\tau_B}$. De plus, puisque nous avons choisi un point sous “ MA ” qui est localisé dans la direction orthogonale à “ MB ”, nous pouvons faire l’approximation que $A_{\tau_A} | B_{\tau_B-d\tau} \approx A_{\tau_A} | B_{\tau_B}$. Nous obtenons :

$$G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A_{\tau_A}), h) = G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h) \frac{G_1^{\text{dist}}(h_0, A_{\tau_A} | \bar{B}_{\tau_B}, h)}{G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B}, h)}, \quad (6.11)$$

où $G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h)$ est la fonction monostatique de Smith de l’équation (6.3), qui ne tient pas compte des événements associés au rayon “ MA ”.

$G_1^{\text{dist}}(A_{\tau_A} | \bar{B}_{\tau_B}, h)$ est la probabilité que la surface soit sous le rayon “ MA ” à τ_A – la hauteur de la surface est dans l’intervalle $]-\infty, h_{\tau_A}[$ sous A_{τ_A} – sachant que le rayon “ MB ” intersecte la surface à τ_B – la hauteur de la surface est exactement $h_{\tau_B} = h_0 + \tau_B \mu_B$. L’information donnée par le point B_{τ_B} donne

de l'information sur la distribution des hauteurs de la surface sous A_{τ_A} . La PDF des hauteurs $P^1(h)$ en ce point est transformée en une PDF conditionnelle $P^1(h | h_{\tau_B})$. En supposant que $P^1(h | h_{\tau_B})$ est connue pour une hauteur donnée h_{τ_B} , alors :

$$\begin{aligned} G_1^{\text{dist}}(A_{\tau_A} | \bar{B}_{\tau_B}, h) &= \int_{-\infty}^{h_{\tau_A}} P^1(h | h_{\tau_B}) dh \\ &= C^1(h_{\tau_A} | h_{\tau_B}). \end{aligned} \quad (6.12)$$

$G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B}, h)$ est la probabilité que la surface soit sous le rayon "MA" à τ_A sachant qu'elle est sous le rayon "MB" à τ_B . L'événement B_{τ_B} fixe les valeurs de hauteur $h_{\tau_B} \in [-\infty, h + \tau_B \mu_B[$ pondérées par la PDF P^1 , ce qui s'écrit :

$$\begin{aligned} G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B}, h) &= \int_{-\infty}^{h + \tau_B \mu_B} \frac{P^1(h_{\tau_B})}{C^1(h + \tau_B \mu_B)} G_1^{\text{dist}}(A_{\tau_A} | \bar{B}_{\tau_B}, h) dh_{\tau_B} \\ &= \int_{-\infty}^{h + \tau_B \mu_B} \frac{P^1(h_{\tau_B})}{C^1(h + \tau_B \mu_B)} C^1(h_{\tau_A} | h_{\tau_B}) dh_{\tau_B}. \end{aligned} \quad (6.13)$$

6.4.2 Dérivation de la distribution conditionnelle des hauteurs

Les équations (6.12) et (6.13) demandent la connaissance de la PDF conditionnelle $P^1(h | h_{\tau_B})$. Par la hauteur h_{τ_B} et la pente \tilde{m} , la hauteur à une distance d peut-être approximée par un développement limité d'ordre 1 de la manière suivante : $h_{\tau_B} + \|\tilde{m}\|d$. Puisque la PDF des pentes suit $\mathcal{N}(0, \sigma_{\tilde{m}}^2)$, la PDF de $h_{\tau_B} + \|\tilde{m}\|d$ suit $\mathcal{N}(h_{\tau_B}, (d\sigma_{\tilde{m}})^2)$. ceci est illustré en figure 6.4. Donc, un point localisé sous B à une distance d et à la hauteur h_{τ_B} peut être interprété comme une mesure sur la surface avec une incertitude modélisée par une erreur qui suit une loi gaussienne.

Si deux mesures indépendantes $x_1 \sim \mathcal{N}(m_1, \sigma_1^2)$ et $x_2 \sim \mathcal{N}(m_2, \sigma_2^2)$ sont fusionnées avec l'estimateur non biaisé de variance minimal [Zhu03], le résultat est $x \sim \mathcal{N}(m, \sigma^2)$ avec les paramètres :

$$\begin{aligned} m &= (m_1/\sigma_1^2 + m_2/\sigma_2^2)/(1/\sigma_1^2 + 1/\sigma_2^2), \\ \sigma^2 &= 1/(1/\sigma_1^2 + 1/\sigma_2^2). \end{aligned}$$

Alors, en supposant que $h \sim \mathcal{N}(0, \sigma_h^2)$ et $h_{\tau_B} + \|\tilde{m}\|d \sim \mathcal{N}(h_{\tau_B}, (d\sigma_{\tilde{m}})^2)$ sont des mesures indépendantes, la PDF conditionnelle $P^1(h_{\tau_A} | h_{\tau_B})$ suit la loi $\mathcal{N}(m_{h_{\tau_A}|h_{\tau_B}}, \sigma_{h_{\tau_A}|h_{\tau_B}}^2)$ qui a pour moyenne :

$$m_{h_{\tau_A}|h_{\tau_B}} = \frac{h_{\tau_B}}{1 + \left(\frac{d\sigma_{\tilde{m}}}{\sigma_h}\right)^2}, \quad (6.14)$$

et pour variance :

$$\sigma_{h_{\tau_A}|h_{\tau_B}}^2 = \frac{(\sigma_h d\sigma_{\tilde{m}})^2}{\sigma_h^2 + (d\sigma_{\tilde{m}})^2}. \quad (6.15)$$

La CDF associée est utilisée dans les équations (6.12) et (6.13) et a pour forme :

$$C^1(h_{\tau_A} | h_{\tau_B}) = 1 - \frac{1}{2} \operatorname{erfc}\left(\frac{h_{\tau_A} - m_{h_{\tau_A}|h_{\tau_B}}}{\sqrt{2} \sigma_{h_{\tau_A}|h_{\tau_B}}}\right) \quad (6.16)$$

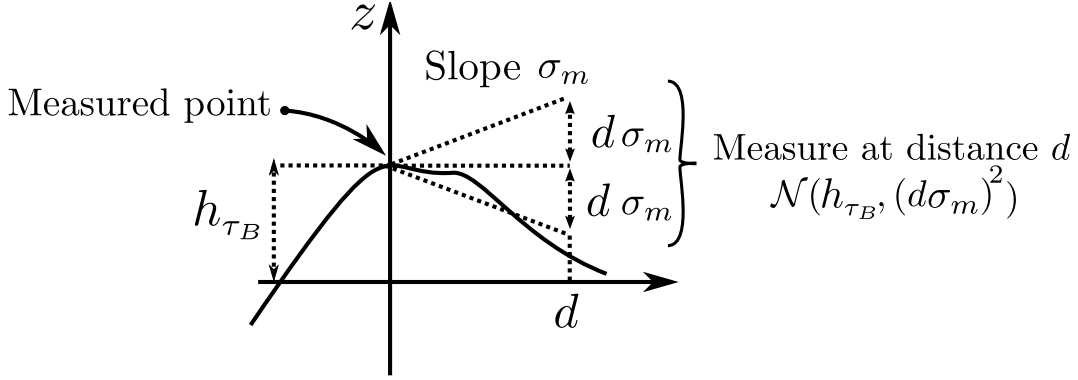


FIGURE 6.4 – La mesure d’un point de la micro-surface apporte de l’information sur son voisinage.

6.4.3 Propriétés de la corrélation azimuthale dans la fonction de masquage et d’ombrage distante

Notre modélisation de la corrélation azimuthale dans le terme $G_2^{\text{dist}}(A \cap B, h)$ est cohérente avec les deux cas extrêmes pour lesquels le résultat est connu :

- Cas sans corrélation $\phi \in [\frac{\pi}{2}, \pi]$.
Nous avons expliqué pourquoi l’information donnée par A n’est pas utilisée lorsque $\phi \in [\frac{\pi}{2}, \pi]$. Dans ce cas, notre modèle revient à la forme séparable sans corrélation $G_2^{\text{dist}}(A \cap B, h) = G_1^{\text{dist}}(A, h) G_1^{\text{dist}}(B, h)$.
- Continuité vers le cas décorréolé $\phi \rightarrow \frac{\pi}{2}$.
Dans l’équation (6.13), la distance entre les points A_{τ_A} et B_{τ_B} est définie par $d = \tau_B \tan \phi$ et $\lim_{\phi \rightarrow \frac{\pi}{2}} d = \infty$. Donc, la variance de la seconde mesure définie en section 6.4.2 tend vers l’infini $\lim_{\phi \rightarrow \frac{\pi}{2}} (\sigma_m d)^2 = \infty$, et l’estimation $\lim_{\phi \rightarrow \frac{\pi}{2}} \mathcal{N}(m_{h_{\tau_A} | h_{\tau_B}}, \sigma_{h_{\tau_A} | h_{\tau_B}}^2) = \mathcal{N}(0, \sigma_h^2)$ tend vers la distribution de la surface. Dans ce cas, aucune information n’est donnée par le second rayon, et $\lim_{\phi \rightarrow \frac{\pi}{2}} G_1^{\text{dist}}(A_{\tau_A} | \bar{B}_{\tau_B}, h) = \lim_{\phi \rightarrow \frac{\pi}{2}} G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B}, h) = P^1(h_{\tau_A})$, et $\lim_{\phi \rightarrow \frac{\pi}{2}} G_1^{\text{dist}}(\bar{B}_{\tau_B} | (B_{\tau_B-d\tau} \cap A_{\tau_A}), h) = G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B-d\tau}, h)$.
Donc, $\lim_{\phi \rightarrow \frac{\pi}{2}} G_1^{\text{dist}}(B | A, h) = G_1^{\text{dist}}(B, h)$, et le modèle tend vers le modèle séparable sans corrélation lorsque la différence azimuthale tend vers $\frac{\pi}{2}$.
- Cas d’une configuration planaire $\phi = 0$.
Dans ce cas, les points sous A sont identiques aux points sous B . Ils partagent la même paramétrisation $\tau = \tau_A = \tau_B$, la distance au voisin le plus proche est $d = 0$ et la PDF $P^1(h_{\tau_A} | h_{\tau_B})$ est un Dirac $\delta_{h_{\tau_B}}$. La probabilité $G_1^{\text{dist}}(A_{\tau_A} | \bar{B}_{\tau_B}, h)$ de l’équation (6.11) devient :

$$\begin{aligned}
 G_1^{\text{dist}}(A_{\tau_A} | \bar{B}_{\tau_B}, h) &= C^1(h_{\tau_A} | h_{\tau_B}) \\
 &= \int_{-\infty}^{h_{\tau_A}} \delta_{h_{\tau_B}}(h') dh' \\
 &= 0,
 \end{aligned} \tag{6.17}$$

puisque A et B sont choisies de telle manière que $\mu_A \leq \mu_B$, i.e., la valeur h_{τ_B} est supérieure à h_{τ_A} et le Dirac vaut 0. Puisque $G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B}, h)$ est la moyenne de $P^1(h_{\tau_A} | h_{\tau_B})$ dans $]-\infty, h_{\tau_B}]$, il est strictement négatif. Donc, l’équation (6.11) vaut 0. La probabilité conditionnelle vaut $G_1^{\text{dist}}(B | A, h) = \exp(0) = 1$, et la probabilité de l’intersection est $G_2^{\text{dist}}(A \cap B, h) = G_1^{\text{dist}}(A, h)$. C’est le résultat attendu d’une configuration planaire où seule la direction d’observation la plus basse est considérée.

6.4.4 Introduction du coefficient de corrélation r_0

En utilisant le résultat de l'équation (6.11) dans l'équation (6.10), nous obtenons :

$$G_1^{\text{dist}}(B | A,) = \exp \left[- \int_0^\infty G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B - d\tau}, h) \frac{G_1^{\text{dist}}(A_{\tau_A} | \bar{B}_{\tau_B}, h)}{G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B}, h)} d\tau_B \right].$$

Pour simplifier l'expression nous introduisons un coefficient de corrélation :

$$r(\tau_B) = \frac{G_1^{\text{dist}}(A_{\tau_A} | \bar{B}_{\tau_B}, h)}{G_1^{\text{dist}}(A_{\tau_A} | B_{\tau_B}, h)}, \quad (6.18)$$

et nous écrivons la probabilité :

$$\begin{aligned} G_1^{\text{dist}}(B | A, h) &= \exp \left[- \int_0^\infty G_1^{\text{dist}}(\bar{B}_{\tau_B} | B_{\tau_B - d\tau}, h) r(\tau_B) d\tau_B \right] \\ &= \exp \left[-\Lambda(\mu_B) \mu_B \int_0^\infty \frac{P^1(h + \tau_B \mu_B)}{C^1(h + \tau_B \mu_B)} r(\tau_B) d\tau_B \right] \\ &= \exp \left[-\Lambda(\mu_B) \mu_B r_0 \int_0^\infty \frac{P^1(h + \tau_B \mu_B)}{C^1(h + \tau_B \mu_B)} d\tau_B \right], \end{aligned} \quad (6.19)$$

où le coefficient de corrélation moyen est donné par :

$$r_0 = \int_0^\infty \frac{P^1(h + \tau_B \mu_B)}{C^1(h + \tau_B \mu_B)} r(\tau_B) d\tau_B \Bigg/ \int_0^\infty \frac{P^1(h + \tau_B \mu_B)}{C^1(h + \tau_B \mu_B)} d\tau_B. \quad (6.20)$$

L'introduction d'un coefficient de corrélation moyen nous permet d'intégrer l'équation (6.19) similairement à l'équation (6.4) par :

$$G_1^{\text{dist}}(B | A, h) = C^1(h)^{r_0 \Lambda(\mu_B)} \quad (6.21)$$

Nous avons maintenant un modèle évaluable, dont tous les termes sont définis. En revanche, l'évaluation de r_0 nécessite de calculer l'intégrale de l'équation (6.20), ce qui n'est pas pratique. Dans la section suivante, nous proposons une approximation analytique de r_0 .

6.4.5 Approximation pratique de r_0

Le coefficient de corrélation r_0 résulte d'une intégration qui n'a pas de solution analytique. Dans cette section, nous proposons une approximation analytique. En développant r donné par l'équation (6.18) avec la définition de probabilité des équations (6.12) et (6.13), nous réécrivons l'équation (6.20) par :

$$r_0 = \int_0^\infty \frac{P^1(h + \tau_B \mu_B) C^1(h_{\tau_A} | h_{\tau_B})}{\int_{-\infty}^{h + \mu_B \tau_B} P^1(h_{\tau_B}) C^1(h_{\tau_A} | h_{\tau_B}) dh_{\tau_B}} d\tau_B \Bigg/ \int_0^\infty \frac{P^1(h + \tau_B \mu_B)}{C^1(h + \tau_B \mu_B)} d\tau_B, \quad (6.22)$$

où $C^1(h_{\tau_A} | h_{\tau_B})$ est la CDF donnée par l'équation (6.16). Le coefficient de corrélation moyen est une fonction à plusieurs variables $r_0(\phi, h, \mu_A, \mu_B, \sigma_{\bar{m}})$ qu'il est difficile d'approximer directement en raison du nombre important des ses paramètres.

Méthodologie pour trouver une approximation. Nous utilisons plusieurs propriétés pour simplifier cette fonction :

- Les propriétés de notre modèle nous donnent que $r_0(\phi = 0) = 0$ et $r_0(\phi = \pi/2) = 1$.

- En calculant r_0 numériquement avec l'équation (6.22) donnée par notre modèle, nous observons empiriquement que sa sensibilité à la variable h est faible. Nous pouvons donc éliminer la variable h en fixant par exemple $h = 0$ (ou n'importe quelle autre valeur).
- En utilisant l'invariance d'étirement de la section 4.5, nous obtenons :

$$r_0(\phi, h, \mu_A, \mu_B, \sigma_{\tilde{m}}) = r_0(\phi, h, \nu_A, \nu_B, 1/\sqrt{2}),$$

où $\nu_A = \mu_A/\sqrt{2}\sigma_{\tilde{m}}$ et $\nu_B = \mu_B/\sqrt{2}\sigma_{\tilde{m}}$ sont les pentes normalisées.

- De plus, nous observons aussi que la sensibilité aux variables ν_A et ν_B dépend principalement de la différence $\nu_B - \nu_A$.

En approximant $h = 0$ et en remplaçant les paramètres ν_A et ν_B par $\nu_B - \nu_A$, nous obtenons une fonction de seulement deux paramètres pour laquelle nous proposons une approximation analytique :

$$r_0(\phi, \nu_B - \nu_A) = \begin{cases} \log(1 + \beta_1 \phi_2^\beta) / \log\left(1 + \beta_1 (\pi/2)_2^\beta\right) & \text{si } \phi \in [0, \frac{\pi}{2}[\\ 1 & \text{si } \phi \in [\frac{\pi}{2}, \pi] \end{cases} \quad (6.23)$$

où nous avons trouvé les valeurs numériques $\beta_1 = 0.17 / \left(\frac{\mu_B - \mu_A}{\sigma_{\tilde{m}}}\right)^{10.49}$ et $\beta_2 = 8.85$ avec un algorithme d'optimisation non linéaire qui minimise l'erreur sur la fonction de masquage et d'ombrage distante $G_2^{\text{dist}}(A \cap B, h) = G_1^{\text{dist}}(A, h) G_1^{\text{dist}}(B | A, h)$.

Validation de l'approximation. Nous comparons la différence entre notre approximation de r_0 et les résultats obtenus par intégration numérique de r_0 de l'équation (6.22) en figure 6.5. Notre approximation a une erreur maximale de 1.1% sur la fonction de masquage et d'ombrage. De plus, puisque r_0 ne dépend pas des hauteurs de la surface, nous pouvons intégrer sur les hauteurs pour obtenir la fonction de masquage moyennée sur les hauteurs :

$$\bar{G}^{\text{dist}}(B | A) = \int_{-\infty}^{\infty} C^1(h)^{r_0 \Lambda(\mu_B)} P^1(h) dh \quad (6.24)$$

$$= \frac{1}{1 + r_0 \Lambda(\mu_B)} \quad (6.25)$$

La figure 6.6 illustre l'effet du facteur r_0 dans la fonction de masquage et d'ombrage pour différentes valeurs de μ_A et $\sigma_{\tilde{m}}$.

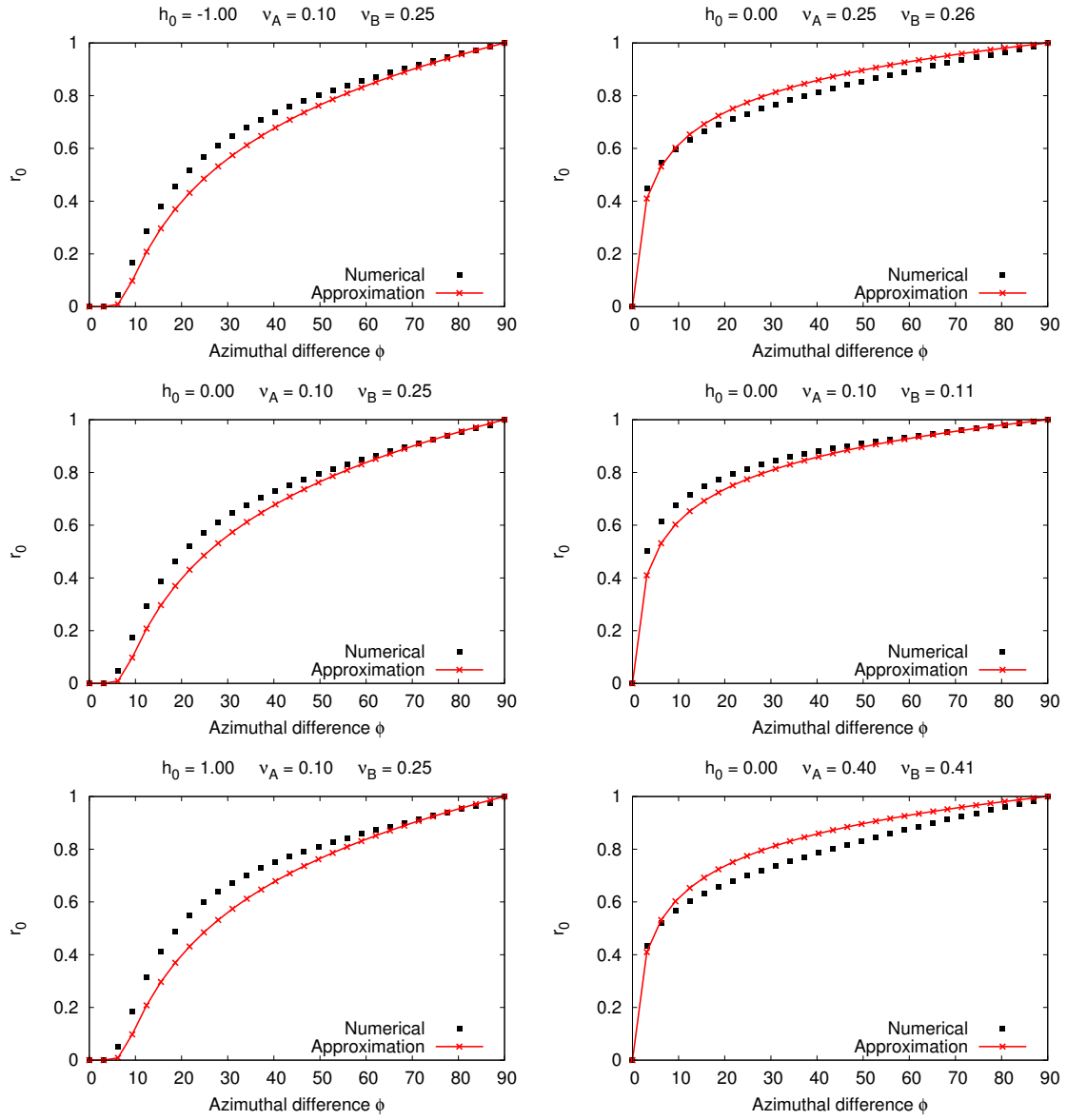


FIGURE 6.5 – Notre approximation analytique du facteur r_0 comparé avec sa valeur de référence pour différentes configurations. Les variations de la hauteur normalisée $h_0 = h/\sqrt{2}\sigma_h$ ont un impact faible sur r_0 (a, c, e). Ce facteur dépend principalement de $\nu_B - \nu_A$, qui est égal à 0.15 et 0.01 respectivement dans (a,c,e) et (b,d,f).

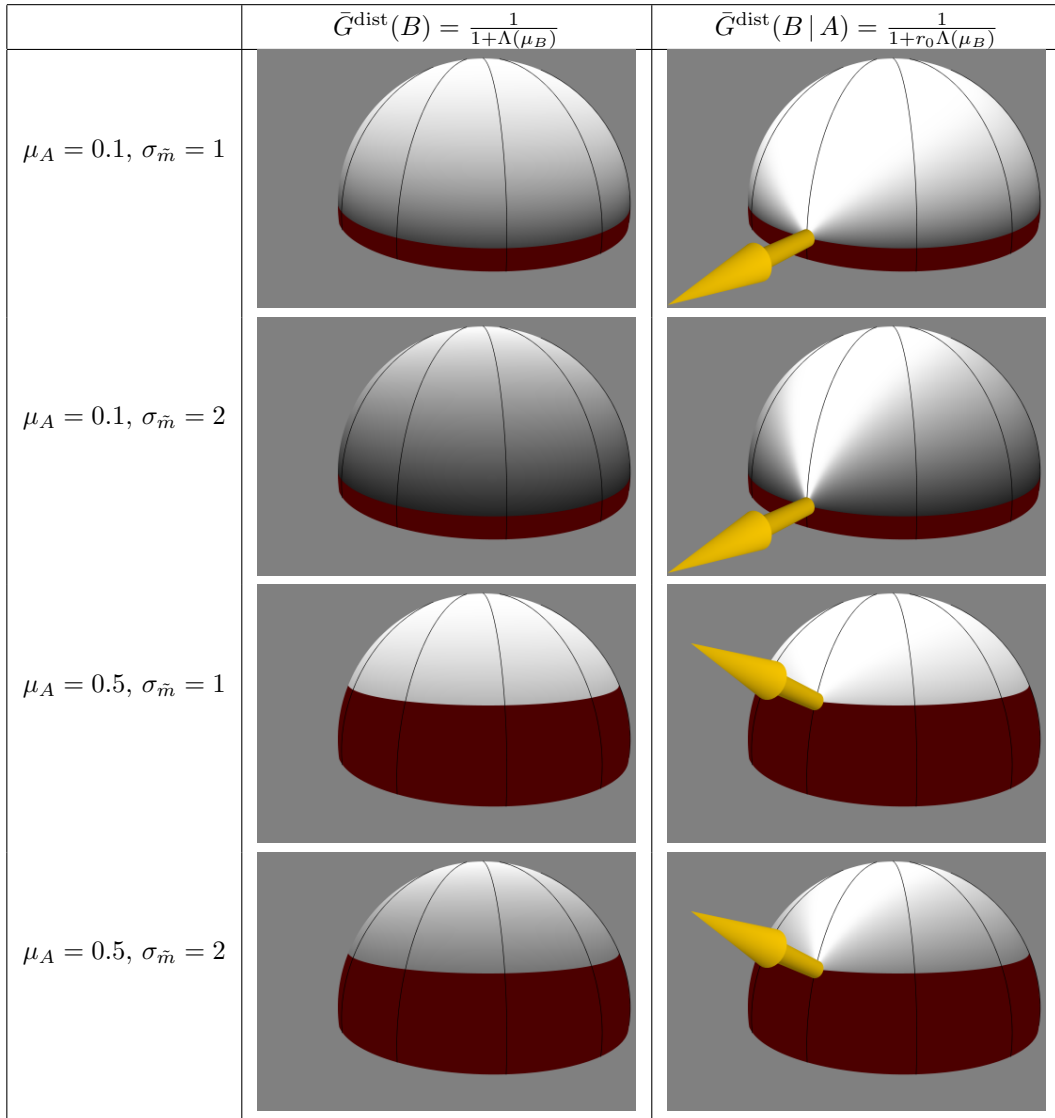


FIGURE 6.6 – Illustration de la probabilité conditionnelle $\bar{G}^{dist}(B|A)$ que la direction B ne soit pas masquée sachant que celle de A ne l'est pas, pour différentes valeurs de μ_A et $\sigma_{\bar{m}}$. La probabilité pour la direction B est affichée en niveau de gris (blanc = 1 et noir = 0) et la direction de A est donnée par la flèche. (gauche) Le modèle standard $\bar{G}^{dist}(B|A) = \bar{G}^{dist}(B)$ ne tient pas compte de l'information introduite par A et la probabilité de masquage ne dépend pas de la différence azimuthale entre A et B . (droite) Notre modèle utilise l'information apportée par la direction A . Plus la direction de B est proche de celle de A , plus la probabilité de visibilité augmente pour la direction B . Dans la zone rouge où $\mu_B < \mu_A$, les modèles ne sont pas définis (ils sont définis seulement pour $\mu_A < \mu_B$).

6.5 Expression finale du modèle proposé

En résumé, pour une surface 2D et des directions d'observation A et B telles que $\mu_A < \mu_B$, nous obtenons les formes suivantes.

Fonction de masquage et d'ombrage par hauteur et par pente. La fonction de masquage et d'ombrage par hauteur et par pente est donnée par :

$$G_2(A \cap B, M = (h, \tilde{m})) = G_2^{\text{local}}(A \cap B, \tilde{m}) G_1^{\text{dist}}(A, h) G_1^{\text{dist}}(B | A, h),$$

où la probabilité $G_1^{\text{dist}}(A, h) = C^1(h)^{\Lambda(A)}$ est définie par l'équation (6.4) ($\theta \rightarrow \theta_A$) et la probabilité $G_1^{\text{dist}}(B | A, h) = C^1(h)^{r_0 \Lambda(\mu_B)}$. L'intégration sur les pentes de $G_2^{\text{local}}(A \cap B, \tilde{m})$ est donnée par l'équation 6.8.

Fonction de masquage et d'ombrage par pente moyennée sur les hauteurs. La forme moyennée sur les pentes est donnée par :

$$\bar{G}_2(A \cap B, h) = \bar{G}_2^{\text{local}}(A \cap B) G_1^{\text{dist}}(A, h) G_1^{\text{dist}}(B | A, h).$$

Puisque r_0 ne dépend pas des hauteurs, nous pouvons calculer la forme moyennée sur les hauteurs :

$$\begin{aligned} \bar{G}_2(A \cap B, \tilde{m}) &= G_2^{\text{local}}(A \cap B, \tilde{m}) \int_{-\infty}^{\infty} G_1^{\text{dist}}(A, h) G_1^{\text{dist}}(B | A, h) P^1(h) dh \\ &= G_2^{\text{local}}(A \cap B, \tilde{m}) \int_{-\infty}^{\infty} C^1(h)^{\Lambda(\mu_A) + r_0 \Lambda(\mu_B)} P^1(h) dh \\ &= \frac{G_2^{\text{local}}(A \cap B, \tilde{m})}{1 + \Lambda(\mu_A) + r_0 \Lambda(\mu_B)}. \end{aligned}$$

C'est celle-ci qui est à utiliser dans les modèles de BRDF. En effet, nous avons vu dans la section 4.4.1 que, puisque dans le modèle les pentes et les hauteurs sont supposées être indépendantes et que seules les pentes (les normales) interviennent dans la BRDF, il faut moyenner la fonction de masquage sur les hauteurs, mais pas sur les pentes qui sont traitées individuellement dans la BRDF.

Fonction de masquage et d'ombrage moyennée sur les hauteurs et les pentes. La forme moyennée sur les hauteurs et les pentes est donnée par :

$$\bar{\bar{G}}_2(A \cap B) = \frac{\bar{G}_2^{\text{local}}(A \cap B)}{1 + \Lambda(\mu_A) + r_0 \Lambda(\mu_B)}.$$

Si cette forme ne sera jamais utilisée dans les autres chapitres, nous la donnons ici car nous nous en servons dans la section 10.3 qui propose des comparaisons avec des données mesurées. En effet, dans les données mesurées, il n'est pas possible de séparer le masquage local du masquage distant, car le masquage est mesuré en moyenne et de manière globale sur l'ensemble de la micro-surface. Nous ne pouvons donc comparer les données mesurées qu'à travers une mesure équivalente donnée par le modèle, qui est cette forme moyennée sur à la fois les hauteurs et les pentes, c'est-à-dire la moyenne globale sur la micro-surface.

6.6 Résultats

La figure 6.7 présente des comparaisons entre des fonctions de masquage bistatiques mesurées et les différents modèles de $\bar{G}_2(A \cap B)$. Nous comparons les données suivantes :

- *Monte Carlo* : nous utilisons 3000 échantillons par configuration pour intégrer la fonction de masquage bistatique sur un échantillon de surface généré avec une fonction de bruit procédurale à statistiques gaussiennes à fonction d'auto-corrélation gaussienne.
- *Bistatic 1-D* est le résultat calculé lorsque A et B sont dans le même plan, où :

$$\bar{G}_2(A \cap B) = \bar{G}_1(A).$$

Ce résultat est constant dans la figure 6.7, puisqu'il ne dépend pas de la différence azimuthale ϕ . C'est la borne supérieure de la fonction de masquage et d'ombrage, car il ne tient pas compte de l'ombrage (l'ombrage vaut 1 et est donc toujours au maximum).

- *Bistatic 2-D uncorrelated* est le modèle séparable où

$$\bar{G}_2(A \cap B) = \bar{G}_1(A) \bar{G}_1(B).$$

Ce résultat est constant dans la figure 6.7 puisqu'il ne dépend pas de la variable ϕ . C'est la borne inférieure de la fonction de masquage et d'ombrage, car il ne tient compte d'aucune corrélation entre les deux (l'ombrage est donc toujours au minimum).

- *Bourlier's model* Ce modèle tient compte des corrélations locales, par les pentes, mais ne tient pas compte de la corrélation azimuthale. Il est donnée par :

$$\bar{G}_2(A \cap B) = \bar{G}_2^{\text{local}}(A \cap B) \bar{G}_2^{\text{dist}}(A \cap B) = \frac{\bar{G}_2^{\text{local}}(A \cap B)}{1 + \Lambda(\mu_A) + \Lambda(\mu_B)}.$$

- *Our model* c'est le modèle de Bourlier lorsque $\Lambda(\mu_B)$ est multiplié par notre approximation analytique du coefficient de corrélation r_0 :

$$\bar{G}_2(A \cap B) = \bar{G}_2^{\text{local}}(A \cap B) \bar{G}_2^{\text{dist}}(A \cap B) = \frac{\bar{G}_2^{\text{local}}(A \cap B)}{1 + \Lambda(\mu_A) + r_0 \Lambda(\mu_B)}.$$

Les graphes montrent que l'effet de corrélation azimuthale est important lorsque ϕ est petit et lorsque l'angle d'incidence θ est important. Intuitivement, moins le point est visible, plus l'effet de corrélation est significatif.

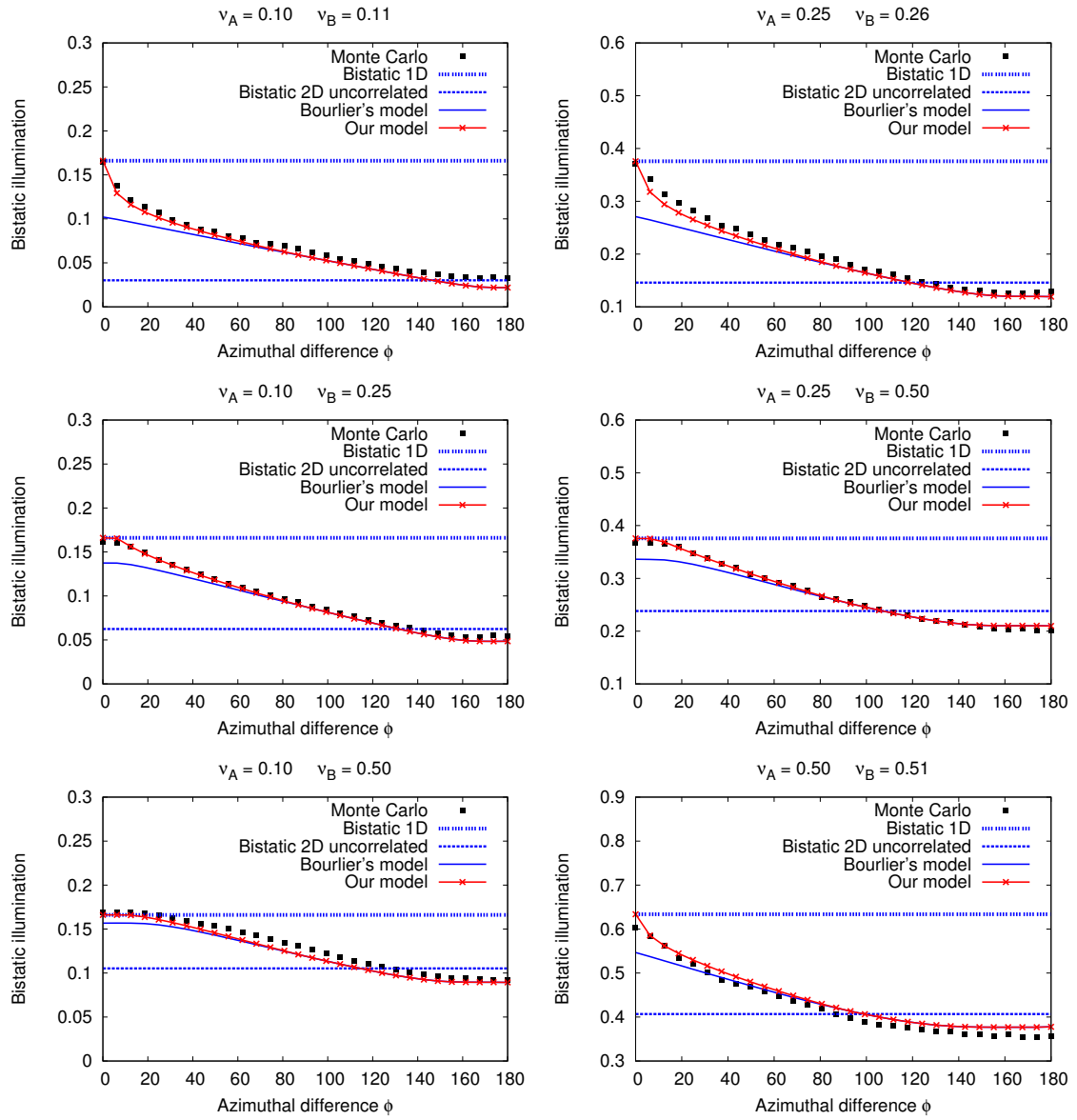


FIGURE 6.7 – *Comparison entre des résultats mesurés sur un échantillon de surface synthétique avec différentes fonctions de masquage bistatiques. Les paramètres de la configuration sont les pentes normalisées : $\nu = \mu/\sqrt{2}\sigma_m$.*

6.7 Résumé du chapitre

Points clés du chapitre

- Les modèles physiques précédents pour la fonction de masquage et d'ombrage ne tiennent pas compte de la corrélation azimuthale entre l'ombrage et le masquage.
- Cet effet est significatif lorsque la rugosité de la micro-surface est importante, lorsque les angles d'incidences sont importants et lorsque la différence azimuthale est faible.
- Nous proposons un modèle dont les prédictions sont précises comparées à des mesures obtenues sur une surface procédurale à statistiques gaussiennes.
- Notre modèle consiste en l'introduction d'un facteur de corrélation r_0 dans la fonction de masquage et d'ombrage de Smith du modèle précédent.
- Nous proposons une approximation analytique de r_0 , rendant le nouveau modèle très facile à utiliser et à implémenter en pratique, en particulier lorsqu'une implémentation de la fonction de masquage et d'ombrage de Smith du modèle précédent est déjà disponible.
- Le nouveau modèle que nous proposons est dérivé pour une surface isotrope centrée à statistiques gaussiennes (une surface de Beckmann isotrope et centrée), mais en utilisant les propriétés d'invariance proposées dans les chapitres précédents, il se généralise trivialement aux micro-surfaces anisotropes et décentrées. Il est donc compatible avec ce que nous proposons dans les chapitres de la partie II, essentiellement basés sur ces micro-surfaces gaussiennes anisotropes et décentrées.

A voir

- L'article associé à ce chapitre, paru dans *Waves in Random and Complex Media* [HBP13].
- L'annexe B qui donne l'implémentation du calcul du facteur r_0 .

Deuxième partie

Représenter et évaluer les BRDFs
multi-échelles

Chapitre 7

Cahier des charges pour une application de rendu multi-échelles

L'objet la partie II est de proposer des manières concrètes de représenter et de calculer l'apparence des surfaces multi-échelles. Nous serons donc amenés à faire des choix techniques sur les représentations et les algorithmes à mettre en oeuvre. Dans ce chapitre, nous donnons les critères qui guident les choix faits dans les chapitres suivants.

7.1 La contrainte de simplicité

Il existe souvent plusieurs méthodes pour résoudre un problème et certains choix doivent être faits de manière subjective ou arbitraire à un moment donné. Une des composantes principales de mon cahier des charges personnel et subjectif est la *simplicité*.

Effectivement, je n'aime pas les algorithmes compliqués : je n'ai aucune envie de les implémenter et encore moins de passer du temps à les débayer. J'ai consacré une partie importante de ma thèse à travailler sur des modèles théoriques, mais j'ai aussi passé beaucoup de temps à mettre au point des algorithmes qui ont pour propriété principale d'être simples. Le mot "simple" est souvent perçu de manière péjorative. Par exemple, on hésite à publier une idée lorsqu'elle semble trop simple. Mais "simple" ne doit pas être confondu avec "trivial". Si trivial n'est pas intéressant, simple me semble au contraire être une bonne propriété essentielle pour un résultat, un algorithme ou une représentation. Ci-dessous quelques arguments pour appuyer ce critère.

Robustesse. Les algorithmes les plus simples ont moins de chances de se retrouver dans des cas particuliers sources de bugs.

Développement, maintenance et optimisation. Plus un algorithme est simple, plus il est facile à développer, débayer, maintenir et à optimiser.

Efficacité. Les algorithmes les plus simples sont souvent les plus efficaces, en particulier sur les architectures SIMD (celles des GPUs) qui supportent mal les cas particuliers. Une approche avec du code simple est parfois plus payante qu'une approche essayant d'optimiser des cas particuliers en introduisant des tests et des opérations supplémentaires. Par ailleurs, un code simple est plus facilement parallélisable.

Reproductibilité et impact. Je ne suis pas le seul à ne pas avoir envie de coder des algorithmes compliqués. Plus un algorithme est compliqué, plus il coûte cher en ressources humaines et moins il a de chances d'être utilisé. Un algorithme simple et facile à reproduire a beaucoup plus de chances d'avoir de l'impact dans la recherche académique comme dans l'industrie.

7.2 Contraintes mémoire

Dans cette section, nous décrivons les contraintes associées au stockage des données en mémoire.

Légereté de la représentation. La représentation doit être raisonnable en terme de coût mémoire. Mais le domaine du raisonnable est difficile à définir dans l'absolu. Par contre, il peut-être défini de manière relative aux utilisateurs visés par l'application que nous proposons. Ces utilisateurs ont probablement déjà leur propre représentation pour laquelle ils ont alloué un certain budget mémoire. Idéalement, la nouvelle représentation ne coûte pas plus cher en espace mémoire que la représentation existante. Lorsque ce n'est pas le cas, il faut étudier la rentabilité d'ajouter de nouvelles choses. Que pouvons-nous faire de plus avec chaque nouvel octet que nous nous autorisons ? En effet, comme nous en avons discuté dans l'introduction, si les puissances et les mémoires des machines augmentent, la quantité du contenu augmente encore plus vite. De fait, il est peu probable que des extensions importantes du budget mémoire par *texel* (élément de texture) soient possibles.

A titre d'exemple, dans les jeux vidéos, les surfaces sont représentées avec des textures d'albédo (3 canaux), des textures de normales (2 canaux), un paramètre de rugosité (1 canal, souvent un exposant de Phong), et quelques autres paramètres de matériau, ce qui fait un budget total de 7-10 canaux par texel. Ce genre de budget peut servir de base pour définir ce qui est raisonnable. Par exemple, en partant de ce constat, une représentation avec des harmoniques sphériques qui ne demande pas moins de 64 valeurs par texel [HSRG07], soit environ 8 fois le budget existant, ne peut pas être qualifiée de raisonnable et a très peu de chances d'être utilisée un jour.

Représentation fermée. Faire cohabiter différentes représentations est souvent difficile à gérer et crée des cas particuliers indésirables. Nous préférons utiliser la même représentation à toutes les échelles. La représentation doit donc être un *système fermé* qui unifie les échelles macro-, méso- et microscopiques. Un système fermé signifie qu'à travers l'espace et les échelles, la représentation reste la même, seuls ses paramètres changent. Pour le dire de manière formelle, on souhaite que la représentation possède une sorte de *structure de groupe* : la représentation définit un ensemble de paramètres avec un opérateur de combinaison (pour l'interpolation et le filtrage) qui garantit que la combinaison de plusieurs paramètres produit toujours de nouveaux paramètres à l'intérieur du même ensemble.

Paramètres linéairement combinables. Dans le paragraphe précédent nous avons évoqué la combinaison des paramètres de la représentation. Pour aller plus loin, nous souhaitons que les paramètres soient *linéairement combinables*.

La première forme de combinaison linéaire dont nous souhaitons profiter est l'interpolation linéaire entre les texels, faite en hardware sur le GPU. Comme nous en avons déjà brièvement discuté en section 1.3.2, il est préférable d'interpoler les paramètres de la représentation plutôt que les valeurs de la BRDF après son évaluation. En effet, en interpolant les paramètres de la représentation, la BRDF n'a besoin d'être intégrée qu'une seule fois avec l'éclairage. Au contraire, interpoler la valeur après l'évaluation de la BRDF demande d'intégrer l'éclairage pour chaque texel à interpoler. Avec du MIPmapping (filtrage tri-linéaire qui implique 8 texels) et éventuellement du filtrage anisotrope de texture (entre $2\times$ et $8\times$), cela peut demander entre 16 et 64 intégrations de BRDFs, ce qui n'est pas acceptable.

La seconde forme de combinaison linéaire qui nous intéresse est liée à la construction de la représentation multi-échelles. En effet, une autre bonne propriété lorsque la représentation est linéairement combinable est que la construction de la représentation sur plusieurs échelles est triviale : les paramètres de la représentation à une échelle donnée sont simplement la moyenne des paramètres à l'échelle inférieure, ce qui rend l'initialisation de la représentation très facile à faire (un simple MIPmapping hardware suffit).

7.3 Contraintes pour la représentation

Dans cette section, nous décrivons les contraintes associées à la précision de la représentation pour la reconstruction de la BRDF.

Compatible avec l'existant. Une nouvelle représentation qui impose aux utilisateurs de jeter toutes leurs données existantes et de remettre en question leurs outils, voire leur paradigme complet de création de contenu, a peu de chances d'avoir du succès. La nouvelle représentation est supposée simplifier la vie de ses utilisateurs et non leur imposer des contraintes supplémentaires. De fait, elle doit être compatible au maximum avec les données et les processus existants.

Qualité minimale. La moindre des choses est d'avoir un résultat exact dans les cas les plus simples lorsqu'il n'y a pas de filtrage (par exemple de près ou avec des height maps parfaitement plates) : il n'est en aucun cas acceptable que l'algorithme de rendu filtré puisse être pire qu'un algorithme naïf sans filtrage.

Robustesse. La représentation doit être robuste dans le sens où elle doit toujours être bien définie. L'espace de ses paramètres ne doit pas contenir certaines valeurs qui créent des instabilités, des cas particuliers ou des biais systématiques (par exemple avec des directions privilégiées).

Basses fréquences et hautes fréquences. La représentation doit être capable de représenter l'ensemble du spectre des BRDFs, c'est-à-dire avec des basses fréquences mais aussi des hautes fréquences. Cela exclut d'emblée les décompositions dans des bases de fonctions comme les harmoniques sphériques où seules les basses fréquences peuvent être représentées avec un nombre raisonnable de coefficients.

Anisotropie. Idéalement, la représentation est capable de capturer l'anisotropie des détails de la surface filtrée.

Initialisation directe. La représentation pré-filtrée doit être facile et rapide à initialiser à partir des données non filtrées. Idéalement, la représentation s'initialise en une seule passe déterministe sur les données. Typiquement, nous préférons éviter les systèmes d'optimisation non linéaires qui sont compliqués à implémenter, imprédictibles et source de bugs. Comme nous l'avons expliqué plus haut, lorsque les paramètres de la représentation sont linéairement combinables, un simple MIPmapping suffit à initialiser la représentation pré-filtrée. C'est le mode d'initialisation idéal.

Compatibilité avec l'animation. Nous ne souhaitons pas seulement rendre des scènes statiques, mais aussi des scènes animées. En effet, les personnages des scènes d'animation sont parmi les objets des scènes les plus susceptibles d'être modélisés par des surfaces détaillées. Mais une technique qui demanderait le recalcul complet de l'ensemble des textures à chaque image de l'animation ne serait pas efficace. Nous souhaitons donc que la représentation soit construite de manière à être compatible avec l'animation de la géométrie sous-jacente.

Représentations procédurales multi-échelles. Certains shaders de production utilisent des données générées procéduralement. Puisque celles-ci n'existent pas en mémoire, il n'est pas possible d'utiliser une représentation pré-filtrée mais nous souhaitons quand même avoir un rendu multi-échelles. Dans ce cas, l'évaluation procédurale doit être compatible avec l'algorithme de rendu multi-échelles, ce qui signifie que l'évaluation procédurale doit produire à chaque échelle les données macro-, méso- et microscopiques équivalentes à celles qui seraient obtenues avec du pré-filtrage.

7.4 Contraintes pour le calcul

Le choix d'une bonne représentation permet de transformer le problème d'intégration 4D spatial-éclairage en un problème d'intégration 2D, où les deux dimensions restantes sont celles de l'éclairage, qu'il faut savoir intégrer. Nous décrivons les contraintes associées à l'évaluation de la BRDF, c'est-à-dire la partie qui concerne le calcul de l'intégration de l'éclairage effectué au moment du rendu. Nous souhaitons mettre au point des algorithmes d'intégration efficaces pour des applications temps-réel mais aussi pour des applications offline. Si les techniques de rendu diffèrent beaucoup en fonction des contraintes des utilisateurs, il est important que la représentation soit compatible avec différentes techniques de rendu. Par exemple, les sociétés de production de films utilisent essentiellement des moteurs de rendu offline (souvent basés sur du lancer de rayon). Mais ces mêmes sociétés utilisent aussi des logiciels de pré-visualisation, destinés aux artistes, dont les performances doivent être au moins interactives. Cela signifie que les mêmes données sont susceptibles d'être utilisées avec plusieurs méthodes de rendu différentes. C'est l'esprit dans lequel le format P-tex a été inventé [BL08].

Evaluation temps-réel. Aussi bien dans le monde du temps-réel (jeu vidéo) que dans le monde du rendu offline (films d'animation), les éclairages sont complexes et de différents types. En particulier, nous souhaitons que la BRDF soit évaluable en temps-réel lorsque l'éclairage est donné par des sources ponctuelles ou directionnelles, mais aussi lorsqu'il est représenté avec de l'*image based lighting* (éclairage basé image) [Deb05]. Dans ce cas, il n'est pas possible d'obtenir une intégration exacte en temps-réel. Nous souhaitons alors une manière d'ajuster le compromis entre la précision du calcul et les performances. De plus, la technique doit converger vers un résultat exact lorsque le budget de calcul augmente. Une technique de rendu temps-réel avec ces contraintes est présentée dans le chapitre 10.

Evaluation dans un moteur d'illumination globale. Nous souhaitons que la représentation soit compatible avec des techniques d'éclairage global, souvent utilisées dans le rendu offline. Ces techniques sont généralement basées sur des algorithmes de lancer de rayons. Pour évaluer la BRDF efficacement, ces algorithmes utilisent systématiquement de l'*importance sampling*. En conséquence, une BRDF qui ne peut pas être importance samplée efficacement ne peut simplement pas être utilisée en production. Nous souhaitons donc associer une stratégie d'importance sampling efficace à la représentation de la BRDF de manière à ce qu'elle puisse être utilisée dans le paradigme d'un moteur de production offline. C'est l'objet du chapitre 11.

Chapitre 8

Rappels sur les représentations des surfaces multi-échelles

L'objet de ce chapitre est de choisir une représentation multi-échelles pour les surfaces détaillées qui sont représentées par des height maps. Ce chapitre contient essentiellement des rappels de techniques existantes. Dans la section 8.1, nous commençons par rappeler les représentations multi-échelles existantes et nous les discutons relativement au cahier des charges fixé dans le chapitre 7.

Type de représentation	Mémoire	Initialisation	Linéairement combinable	Basses et hautes fréquences	Anisotropie	Multi-modalité	Animation
Mono-lobe	✓	✓	✓	✓	✗	✗	✗
Multi-lobe	✗	✗	✗	✓	✓	✓	✗
Lobes anisotropes	✓	✗	✗	✓	✓	✗	✗
Bases orthogonales	✗	✓	✓	✗	✓	✓	✗
Clustering	✗	✗	✗	✗	✓	✓	✗
LEAN Mapping	✓	✓	✓	✓	✓	✗	✓

TABLE 8.1 – Comparaison des représentations multi-échelles de micro-facettes.

Dans la section 8.2, nous expliquons pourquoi notre choix se porte sur la représentation *LEAN Mapping* (Linear Efficient Antialised Normal Mapping) [OB10], basée sur des distributions de pentes à statistiques gaussiennes. Nous rappelons les détails techniques de cette représentation et ses propriétés. Nous verrons dans les chapitres 9 et 10 comment utiliser cette représentation dans une BRDF multi-échelles, nous appellerons cette technique *LEADR Mapping* (Linear Efficient Antialised Displacement and Reflectance Mapping) [DHI⁺13].

La contribution de ce chapitre qui n'est pas un simple rappel est présentée en section 8.3 : nous montrons comment rendre la représentation de LEAN Mapping compatible avec l'animation de la surface géométrique sous-jacente, ce qui permet de l'utiliser sur des modèles animés ou en cours d'édition sans avoir à recalculer toutes les données.

8.1 Rappels généraux sur les représentations multi-échelles

Dans cette section, nous rappelons les différents types de représentations existantes pour les détails des surfaces multi-échelles. En pratique, les données sont stockées dans des textures pré-filtrées représentées avec des niveaux de MIPmap [Wil83]. Lorsque nous utilisons un modèle à micro-facettes multi-échelles, les données des textures servent à représenter les distributions de micro-facettes $D(\omega_m)$ à chaque échelle et dans chaque voisinage \mathcal{M} de la surface. La question est donc de choisir la meilleure représentation pour D . Les manières de la représenter sont de 5 types et leurs propriétés respectives sont illustrées dans la figure 8.1.

8.1.1 Les représentations mono-lobe

La manière la plus simple de représenter une distribution de normales D est de l’approximer par un seul lobe directionnel, donné par sa direction principale et un paramètre de rugosité qui donne la largeur du lobe : ce sont des représentations dites *mono-lobe*. Il en existe plusieurs types : les lobes gaussiens [WRG⁺09], les lobes en puissance de cosinus [MPA⁺10] et la distribution de von Mises [HSRG07].

Le principal avantage de ces représentations est qu’elles sont faciles à stocker en mémoire : il suffit d’interpoler les directions des lobes et leurs paramètres de rugosité. Par ailleurs, elles demandent peu de mémoire, ce qui les rend très utilisées dans le monde du rendu en temps-réel, notamment dans les jeux vidéos.

La méthode de Toksvig [Tok05] permet même de reconstruire un lobe en cosinus à partir de normales MIPmappées : la moyenne des normales donne à la fois la direction du lobe et le paramètre de rugosité du lobe qui est calculé avec la norme de cette moyenne (intuitivement, plus elle est petite, plus la distribution du lobe est large). Ceci donne un moyen simple d’initialiser la représentation (un simple MIPmap des composantes x_m, y_m, z_m des normales suffit) et permet d’économiser le stockage du paramètre de rugosité.

8.1.2 Les représentations multi-lobes

Le principal reproche fait aux représentations mono-lobe est leur incapacité à représenter des distributions anisotropes ou multi-modales. Certaines représentations utilisent plusieurs lobes pour représenter D [Fou92, TLQ⁺05, HSRG07, TLQ⁺08] : ce sont des représentations dites *multi-lobes*. Le coût de la représentation est proportionnel au nombre de lobes utilisés.

Le défaut principal de ces représentations réside dans leur initialisation : elle nécessite un processus d’optimisation non linéaire pour le placement des lobes, ce qui en fait un processus peu robuste et peu prédictible. En effet, des combinaisons très différentes peuvent donner des reconstructions proches. Par ailleurs, pour que la représentation soit interpolable, il faut que les lobes soient alignés correctement d’un texel à l’autre. Calculer l’alignement optimal est un problème intrinsèquement mal posé et les processus d’optimisation dédiés à cette tâche sont souvent instables, dépendent des conditions initiales et produisent souvent des singularités qui résultent en des artefacts visuels [BN12a].

Pour montrer que ce problème est mal posé nous proposons un exemple simple en 2D illustré dans la figure 8.1. Imaginons que deux texels voisins contiennent chacun deux lobes représentés par des vecteurs 2D. Disons que le premier texel contient les vecteurs $(-1, 0)$ et $(1, 0)$ et le second texel contient les vecteurs $(0, -1)$ et $(0, 1)$. Pour interpoler, il faut choisir une correspondance : quel lobe correspond à quel lobe ? Le lobe $(-1, 0)$ correspond-il plutôt à $(0, -1)$ ou à $(0, 1)$? Il n’y a pas de solution privilégiée, ce qui rend le choix pour une solution plutôt qu’une autre très instable. De plus, les deux solutions produisent une forme anisotrope intermédiaire indésirable alors que la forme attendue aurait été isotrope, ou, du moins, sans direction d’anisotropie dominante.

Ce problème est la raison principale, en plus de leur consommation mémoire importante, pour laquelle ces représentations ne sont jamais utilisées en pratique, et c’est pour cette raison que nous faisons le choix de ne pas les utiliser non plus.

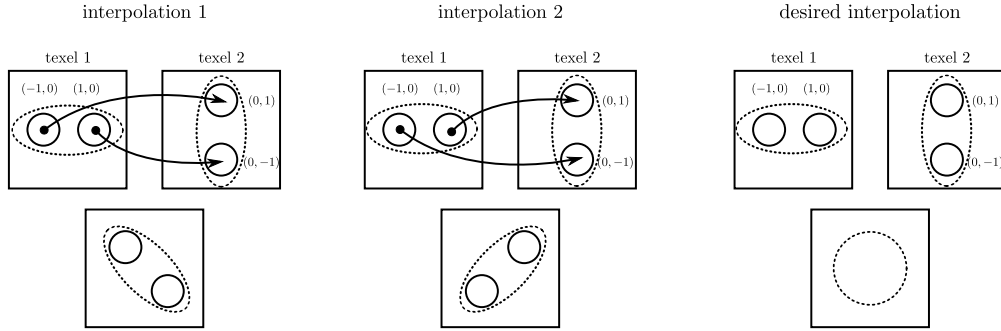


FIGURE 8.1 – L’interpolation multi-lobes est mal posée dans certains cas et produit des artefacts indésirables. Dans cette figure, les deux lobes peuvent être mis en correspondance de deux manières différentes. Les distributions interpolées sont anisotropes avec des directions d’anisotropie différentes, alors que l’interpolation idéale ne devrait pas avoir de direction d’anisotropie privilégiée.

8.1.3 Les représentations à lobes anisotropes

Les lobes dont nous avons parlé jusqu’à présent sont des lobes symétriques par rotation axiale, parfois appelés “isotropes” par abus de langage. Les *anisotropic spherical gaussians* [XSD+13] sont des lobes anisotropes qui sont paramétrés par deux coefficients de rugosité associés à deux directions d’anisotropie.

Le problème majeur de cette représentation par rapport à notre cahier des charges est qu’elle n’est pas linéairement combinable. Le problème d’interpoler les deux directions d’anisotropie est similaire à celui d’interpoler plusieurs lobes : comment faire la correspondance d’un texel à l’autre ?

Le second problème est l’initialisation de la représentation : les auteurs passent par un processus d’optimisation non linéaire pour trouver le lobe anisotrope qui approxime le mieux une distribution cible. Cela est décevant, en particulier pour une distribution qui s’appelle “gaussien”. En effet, une propriété attendue d’une gaussienne est qu’elle soit paramétrée par ses moments, ce qui permet d’obtenir directement une approximation de n’importe quelle distribution en les calculant. Mais les paramètres de ces anisotropic spherical gaussians ne sont pas liés aux moments de la distribution ni à aucun autre descripteur statistique pertinent et calculable.

Une autre faiblesse de cette représentation est qu’aucun des opérateurs (évaluation, produit, convolution) qui lui sont associés ne possède une forme close qui permet de les évaluer simplement et efficacement. Par exemple, pour approximer leur coefficient de normalisation, il faut évaluer deux fonctions de Bessel, pour lesquelles il n’y a pas non plus de forme analytique : elles aussi doivent être approximées ou précalculées. Avant même de vouloir faire quelque chose avec ces distributions, rien qu’évaluer leur coefficient de normalisation est un challenge technique !

Cette idée de lobes anisotropes est donc intéressante a priori mais sa réalisation pratique contient trop de mauvaises propriétés pour qu’elle soit réellement utilisable avec notre cahier des charges.

8.1.4 Les bases de fonctions

Une autre manière de représenter D est de la décomposer dans une base de fonctions orthogonales. Une représentation privilégiée pour décomposer les fonctions sphériques sont les harmoniques sphériques [Fou92, HSRG07] ou les harmoniques hémisphériques [McT04, Gre07, HW10].

Il y a deux avantages majeurs à utiliser une décomposition dans une base orthogonale. D’une part, l’initialisation de la représentation est très simple : il suffit de faire une projection des données sur les fonctions de la base. D’autre part, l’interpolation est bien définie : il suffit d’interpoler les coefficients dans la base. Par ailleurs, des propriétés comme l’anisotropie ou la multi-modalité viennent automatiquement avec la représentation sans avoir besoin d’être traitées de manière particulière.

Mais il faut garder à l’esprit que l’efficacité d’une base de fonction dépend de la fonction à représenter.

Ainsi, le choix d’une base doit être fait en fonction de sa capacité à capturer l’essentiel de la quantité d’information présente dans les données avec peu de coefficients. C’est le cas par exemple pour les irradiance maps : ce sont des fonctions basses-fréquences et leurs neuf premiers coefficients d’harmoniques sphériques permettent d’avoir une très bonne précision [RH01], ce qui en fait une base particulièrement bien adaptée à ces fonctions.

Malheureusement, dans le cas général, la fonction à décomposer n’est pas forcément épars (“sparse” en anglais) dans la base orthogonale choisie : elle se décompose avec beaucoup de coefficients non nuls qu’il faut stocker pour garantir une reconstruction précise. Par exemple, lorsque la distribution ne contient qu’une seule normale (la distribution est un dirac), sa décomposition en harmoniques sphériques a une infinité de coefficients non nuls. De fait, les harmoniques sphériques ne sont pas efficaces pour représenter les fonctions hautes-fréquences et il en va de même pour les autres bases de fonctions.

Les bases de fonctions ne sont donc pas adaptées pour capturer des détails de surface lorsque ceux-ci sont potentiellement utilisés avec des BRDFs hautes-fréquences, par exemple spéculaires ou glossy. C’est la raison pour laquelle nous préférons les approches paramétriques, par exemple à base de lobes, qui permettent de représenter des distributions basses- et hautes-fréquences avec le même budget mémoire.

8.1.5 Les approches par clustering dépendant du point de vue

Le principe des approches par clustering dépendant du point de vue est de construire une représentation comme une fonction décomposée en plusieurs clusters choisis ou pondérés en fonction de la direction d’observation. Par exemple, Becker et Max représentent des distributions de normales dont les paramètres varient selon la direction d’observation [BM93]. Le principe de Characteristic Point Map [WDR09, WDR11] est d’identifier des points particuliers pour chaque point de vue et de combiner leurs contributions en fonction du point de vue.

Ces techniques permettent de représenter précisément les corrélations entre les différents attributs de la surface qui sont à l’origine d’effets complexes dépendants du point de vue. En dehors de cet unique avantage, elles ne respectent aucun des critères du cahier des charges que nous avons fixé : la complexité de la construction de ces représentations, leur coût de stockage et le fait qu’elles ne soient pas interpolables les rend complètement inutilisables en pratique.

8.1.6 LEAN Mapping

Plutôt que de représenter la distribution de normales D directement, il est possible, à la place, de représenter la distribution de pentes P^{22} qui lui est associée. En particulier, la représentation de *LEAN Mapping* [OB10] représente P^{22} avec une gaussienne anisotrope dont les moments sont stockés et servent à reconstruire sa matrice de covariance. Cela est donc équivalent à stocker les paramètres d’une distribution de Beckmann décentrée et corrélée, comme discutée en section 4.5.5. C’est une représentation paramétrique efficace à évaluer, qui demande peu de mémoire et qui bénéficie de la paramétrisation de l’espace des pentes, ce qui la rend linéairement combinable.

Par ailleurs, les autres représentations qui stockent essentiellement des distributions de normales ne supportent pas l’animation. En effet, les normales sont des covecteurs de la géométrie et les transformations appliquées sur la géométrie ne peuvent pas être appliquées de manière simple sur les distributions de normales. En revanche, elles peuvent l’être sur les distributions de pentes, faisant de LEAN Mapping la seule représentation compatible avec l’animation de la surface, comme nous le montrons dans la section suivante.

En résumé, bien que le modèle de BRDF présenté dans LEAN Mapping ne soit pas adapté à notre problème (c’est un problème de filtrage de normal maps, pas de height map, et de toute façon la BRDF proposée dans le papier LEAN Mapping est incomplète car elle ne contient pas le jacobien de la transformation normale/réflexion et ne conserve pas l’énergie), la représentation avec des statistiques gaussiennes dans l’espace des pentes est celle qui présente le plus de bonnes propriétés relativement à notre cahier des charges. C’est donc celle sur laquelle nous baserons notre application dans les chapitres suivants.

Il est à noter qu’il existe une version isotrope de LEAN Mapping [Bak11]. Les dérivations que nous proposons par la suite sont aussi valables pour cette représentation simplifiée.

8.2 Rappels et motivation pour les distributions de pentes à statistiques gaussiennes

Dans cette section, nous rappelons la représentation de LEAN Mapping basée sur une distribution gaussienne dans l'espace des pentes et ses propriétés.

8.2.1 Motivation, choix et limitations

Combinaison linéaire de la pente moyenne. Dans le chapitre 5, nous avons vu qu'un modèle multi-échelles implique la représentation de l'échelle mésoscopique, donnée par la pente moyenne $(\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}})$. Comme nous l'avons vu dans la section 5.1, la pente de la méso-surface est simplement la pente moyenne de la micro-surface. Cela signifie que les paramètres de pentes moyennes peuvent être stockés, MIPmappés et interpolés linéairement. Ils sont donc directement stockables dans une texture.

Motivation pour les statistiques gaussiennes. Stocker la pente moyenne, qui donne la position moyenne de la distribution P^{22} , permet de représenter l'échelle mésoscopique. Il reste donc à savoir comment représenter l'échelle microscopique, c'est-à-dire la forme de la distribution P^{22} . Nous choisissons d'utiliser des distributions de pentes gaussiennes, c'est-à-dire des distributions de Beckmann.

D'une part, les statistiques gaussiennes sont une description naturelle et universelle des distributions. En effet, beaucoup de phénomènes naturels produisent automatiquement des statistiques gaussiennes lorsqu'ils mettent en jeu de nombreux éléments, c'est ce que prédit le théorème Centrale Limite. De plus, la plupart des fonctions de bruit procédurales produisent naturellement des statistiques gaussiennes [LLDD09, LD11], ce qui les rend directement compatibles avec cette représentation.

D'autre part, la distribution de Beckmann est une des distributions standard implémentée dans de nombreux moteurs de rendu modernes *physically based*. Les moteurs plus anciens utilisent souvent des paramètres de rugosité correspondant à une distribution de Phong, mais celle-ci s'approche très bien avec une distribution de Beckmann [OB10].

Enfin, la représentation de LEAN Mapping stocke la rugosité sous la forme de moments d'ordre 2 de la distribution (la pente moyenne étant les moments d'ordre 1). Cette représentation est linéairement combinable : elle peut être interpolée, MIPmappée et supporte le filtrage anisotrope des textures.

Choix du profil de surface. Dans le chapitre 4, nous avons vu qu'il existe plusieurs profils de micro-surfaces et qu'il est indispensable d'en choisir un pour déterminer le masquage et l'ombrage se produisant sur la micro-surface. Nous avons vu qu'il existe deux profils privilégiés dans la littérature : celui de Smith et celui à V-cavités. Bien que les deux ne soient pas réalistes, la BRDF obtenue avec le profil de micro-surface de Smith est très proche de celle obtenue avec des données mesurées sur des surfaces continues et plausibles. Nous faisons donc le choix de supposer que le profil de micro-surface associé à la distribution des pentes P^{22} est le profil de Smith. Avec ce choix et celui de supposer une surface à statistiques gaussiennes (Beckmann), la fonction de masquage G_1 et la forme complète de la BRDF à utiliser sont entièrement déterminées.

Limitations de l'espace des pentes. Choisir une représentation en espace des pentes implique certaines limitations intrinsèques. En particulier :

- Nous supposons que la surface multi-échelles est toujours représentable par un champ de hauteurs h .
- Nous ne tenons pas compte de la courbure de la surface macroscopique. Nous reviendrons sur cette hypothèse dans la section 10.4.

8.2.2 Définition de la représentation

Représentation mathématique. Une distribution de pentes gaussiennes est donnée par :

$$P^{22}(\tilde{m}) = \frac{\exp\left(-\frac{1}{2}(\tilde{m} - \mathbb{E}[\tilde{m}])^t \Sigma^{-1} (\tilde{m} - \mathbb{E}[\tilde{m}])\right)}{2\pi\sqrt{|\Sigma|}},$$

où la pente moyenne est donnée par $\mathbb{E}[\tilde{m}] = (\mathbb{E}[x_{\tilde{m}}], y_{\tilde{m}}) = (\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}})$ et la matrice de covariance Σ est telle que :

$$\Sigma = \begin{bmatrix} \sigma_x^2 & c_{xy} \\ c_{xy} & \sigma_y^2 \end{bmatrix} \quad \text{avec} \quad \begin{aligned} \sigma_x^2 &= \mathbb{E}[x_{\tilde{m}}^2] - \mathbb{E}^2[x_{\tilde{m}}] \\ \sigma_y^2 &= \mathbb{E}[y_{\tilde{m}}^2] - \mathbb{E}^2[y_{\tilde{m}}] \\ c_{xy} &= \mathbb{E}[x_{\tilde{m}}y_{\tilde{m}}] - \mathbb{E}[x_{\tilde{m}}]\mathbb{E}[y_{\tilde{m}}] \end{aligned} \quad (8.1)$$

Représentation mémoire. Dans LEAN/LEADR Mapping, les 5 valeurs flottantes $\mathbb{E}[x_{\tilde{m}}]$, $\mathbb{E}[y_{\tilde{m}}]$, $\mathbb{E}[x_{\tilde{m}}^2]$, $\mathbb{E}[y_{\tilde{m}}^2]$ et $\mathbb{E}[x_{\tilde{m}}y_{\tilde{m}}]$ sont stockées dans une texture. La figure 8.2 montre comment sont organisées et utilisées les données dans notre application. Nous stockons les déplacements h et les moments d'ordre 1 et 2 des pentes de la height map dans des textures MIPmappées. Au moment du rendu, la surface géométrique est tessellée et déplacée avec les valeurs de hauteur pré-filtrées. Les statistiques pré-filtrées des pentes de la surface sont utilisées pour calculer une distribution de Beckmann que nous utilisons dans la BRDF. Cette représentation présente de nombreux avantages techniques : elle requiert peu de mémoire, elle est combinable linéairement, elle supporte l'interpolation, le MIPmapping et le filtrage anisotrope des textures, et elle s'initialise automatiquement à travers le MIPmapping.

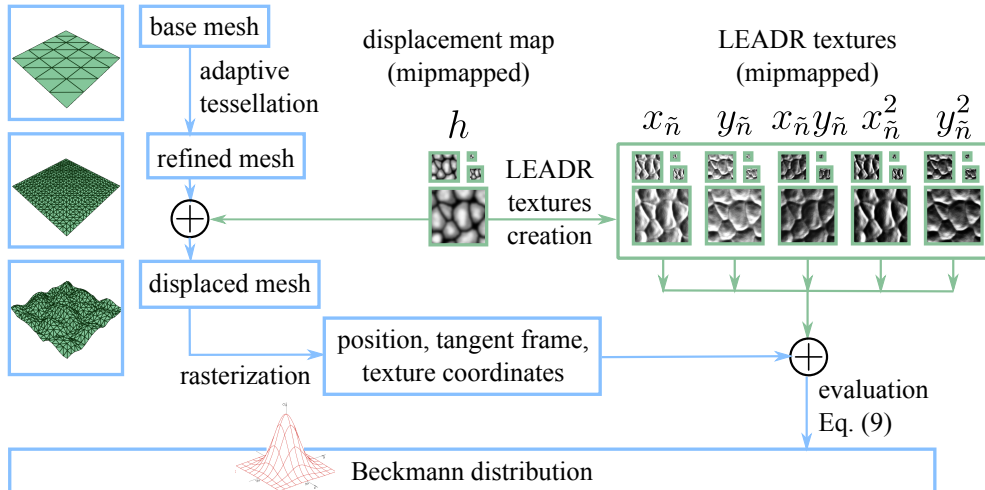


FIGURE 8.2 – Illustration des données de LEAN/LEADR Mapping que nous utilisons pour représenter la surface multi-échelles.

Représentation et évaluation procédurale. Nous avons donné une représentation explicite basée sur les moments de la distribution P^{22} . Il est à noter que certaines représentations procédurales calculent directement les moments d'une distribution gaussienne multi-échelles : c'est le cas du Gabor Noise [LLDD09, LD11]. Bruneton et Neyret utilisent aussi une fonction procédurale pour générer la surface de l'océan animé [BNH10] basée sur la même représentation. Ces fonctions procédurales sont donc complètement compatibles avec ce que nous proposons dans ce chapitre et dans les suivants.

8.3 Compatibilité de la représentation avec l'animation

Problème. Une surface animée et déformable ne conserve pas son apparence, comme le montre la figure 8.3. En effet, l'animation peut étirer ou compresser des zones de la surface et nous avons vu dans la section 4.5 que ces transformations ont pour conséquence de changer la rugosité de la surface et donc sa BRDF. D'un autre côté, nous ne souhaitons pas recalculer toute la représentation pour chaque image d'une séquence d'animation.

Notre solution. Pour rendre la représentation compatible avec les transformations de la géométrie sous-jacente pendant l'animation, nous proposons de stocker les moments de la distribution des pentes en espace texture plutôt que directement en espace objet. Nous utilisons ensuite les jacobiens des déformations pour obtenir la distribution des pentes en espace objet, qui est utilisée dans la BRDF. Calculer les moments de P^{22} de cette manière nous permet d'utiliser les mêmes données pré-calculées sans avoir à les mettre à jour : les modifications de la surface dues à l'animation sont intégrables à la volée directement dans la distribution et l'apparence de la micro-surface modifiée est calculée correctement. Les types de déformations que la représentation supporte de cette manière sont illustrés en figure 8.3.

Représentation en espace texture. La distribution de pentes est calculée en espace texture. Mais les pentes en espace texture ne sont pas nécessairement les pentes en espace objet, en particulier lorsque la paramétrisation de la texture n'est pas parfaite, par exemple lorsque la surface est animée et déformée. En pratique, nous stockons les moments de la distribution des pentes en espace texture paramétré par (u, v) : $\mathbb{E}[u_{\bar{n}}]$, $\mathbb{E}[v_{\bar{n}}]$, $\mathbb{E}[u_{\bar{n}}^2]$, $\mathbb{E}[v_{\bar{n}}^2]$, et $\mathbb{E}[u_{\bar{n}}v_{\bar{n}}]$. Ces moments sont filtrés linéairement au moment du rendu, avec du MIPmapping et éventuellement du filtrage anisotrope. Pour obtenir les moments en espace objet à utiliser pour calculer la BRDF, la distribution doit être exprimée dans le plan tangent (x, y, ω_g) . La paramétrisation de l'espace texture dans le plan tangent est donnée par les fonctions $u(x, y)$, $v(x, y)$. Si nous appelons $h(u, v)$ l'amplitude du déplacement, les pentes en espace texture sont données par le gradient de déplacement :

$$(u_{\bar{n}}, v_{\bar{n}}) = \left(\frac{\partial h}{\partial u}, \frac{\partial h}{\partial v} \right).$$

De fait, les pentes en espace objet sont définies par :

$$(x_{\bar{n}}, y_{\bar{n}}) = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y} \right) = \left(\frac{\partial h}{\partial u} \frac{\partial u}{\partial x} + \frac{\partial h}{\partial v} \frac{\partial v}{\partial x}, \frac{\partial h}{\partial u} \frac{\partial u}{\partial y} + \frac{\partial h}{\partial v} \frac{\partial v}{\partial y} \right).$$

Nous utilisons cette propriété pour calculer les moments de la distribution de pentes en espace objet.

Etirement et torsion de la surface géométrique. Si les distorsions de la paramétrisation de l'espace texture sur la surface sont basses-fréquences, c'est-à-dire qui varient peu localement, nous pouvons raisonnablement supposer que les gradients de la paramétrisation $\frac{\partial u}{\partial x}$, $\frac{\partial v}{\partial x}$, $\frac{\partial u}{\partial y}$, et $\frac{\partial v}{\partial y}$ sont localement constants. Sous cette hypothèse, les moments en espace objet sont donnés par :

$$\begin{aligned} \mathbb{E}[x_{\bar{n}}] &= \frac{\partial u}{\partial x} \mathbb{E}[u_{\bar{n}}] + \frac{\partial v}{\partial x} \mathbb{E}[v_{\bar{n}}] \quad ; \quad \mathbb{E}[y_{\bar{n}}] = \frac{\partial u}{\partial y} \mathbb{E}[u_{\bar{n}}] + \frac{\partial v}{\partial y} \mathbb{E}[v_{\bar{n}}], \\ \sigma_x^2 &= \left(\frac{\partial u}{\partial x} \right)^2 \sigma_u^2 + \left(\frac{\partial v}{\partial x} \right)^2 \sigma_v^2 + 2 \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} c_{uv}, \\ \sigma_y^2 &= \left(\frac{\partial u}{\partial y} \right)^2 \sigma_u^2 + \left(\frac{\partial v}{\partial y} \right)^2 \sigma_v^2 + 2 \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} c_{uv}, \\ c_{xy} &= \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} \sigma_u^2 + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} \sigma_v^2 + \left(\frac{\partial u}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial v}{\partial x} \frac{\partial u}{\partial y} \right) c_{uv}. \end{aligned} \tag{8.2}$$

Changement de l'amplitude du déplacement. Supposons que nous souhaitons modifier l'amplitude des déplacements en multipliant les valeurs de h par un facteur η . Si le facteur η varie peu spatialement et peut être considéré constant localement, alors il suffit de multiplier les moments de premier ordre $\mathbb{E}[x_{\bar{n}}]$ et $\mathbb{E}[y_{\bar{n}}]$ par η , et les moments de second ordre $\mathbb{E}[x_{\bar{n}}^2]$, $\mathbb{E}[y_{\bar{n}}^2]$, et $\mathbb{E}[x_{\bar{n}}y_{\bar{n}}]$ par η^2 .

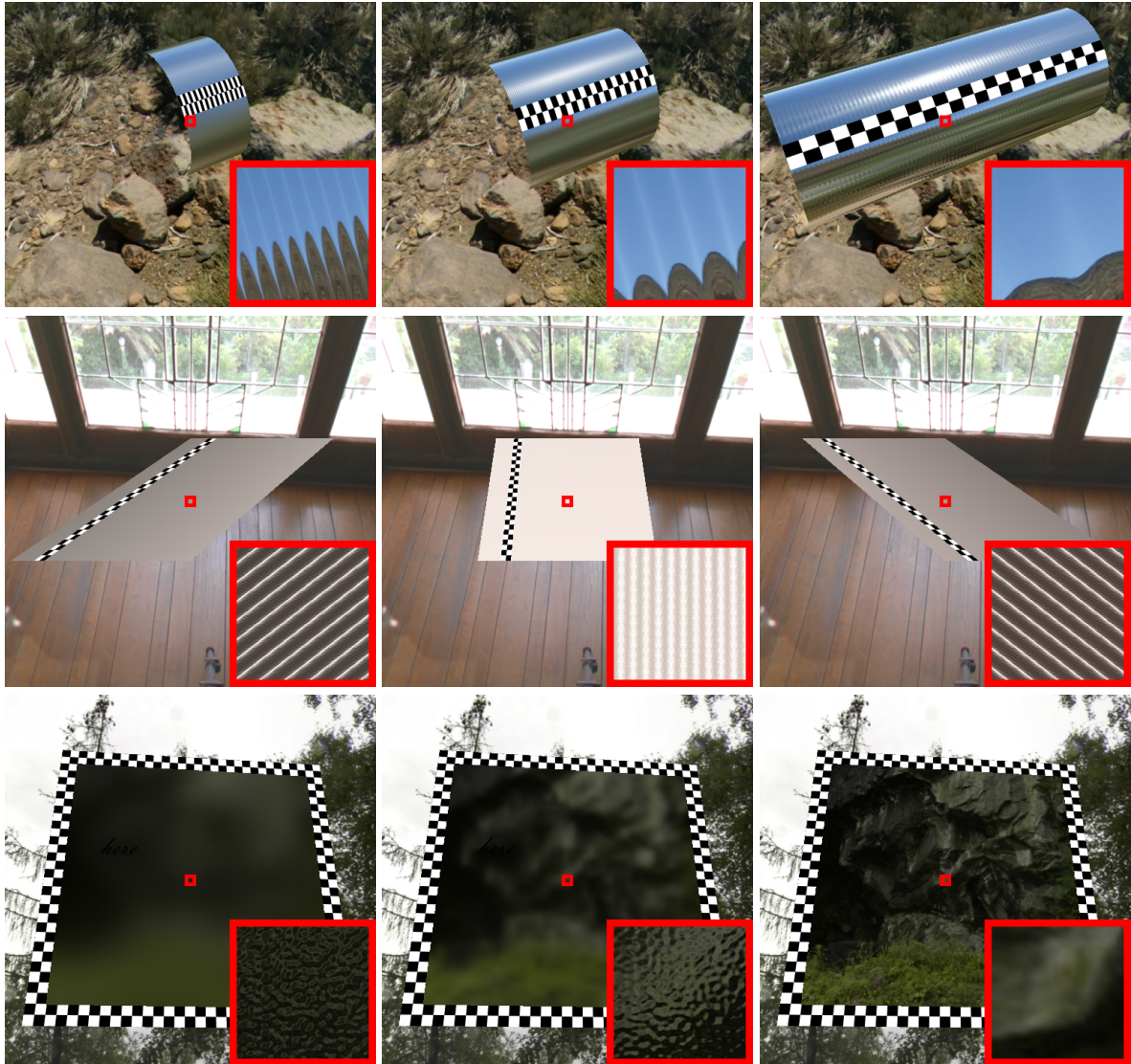


FIGURE 8.3 – (haut) Notre représentation intègre dynamiquement les étirements de la surface : les réflexions spéculaires deviennent anisotropes avec une compression importante de la surface. En effet, étirer la surface réduit l'amplitude des pentes. Puisque notre représentation est linéaire, cet effet peut être introduit à n'importe quelle échelle. (milieu) Une surface avec des micro-facettes diffuses déformées par torsion. La torsion change la direction d'anisotropie et l'apparence de la surface s'en trouve modifiée. De la même manière que l'étirement, cet effet peut être intégré à n'importe quelle échelle. (bas) Réduire l'amplitude de la height map diminue l'amplitude des pentes de la micro-surface : la rugosité décroît et la surface tend à devenir spéculaire.

8.4 Résumé du chapitre

Points clés du chapitre

- Pour représenter la surface multi-échelles, nous choisissons d'utiliser la représentation de *LEAN Mapping* [OB10], une distribution de pentes à statistiques gaussiennes, car c'est celle qui combine le plus de bonnes propriétés par rapport à notre cahier des charges.
- La représentation est légère.
- La représentation est linéairement combinable.
- La représentation permet de représenter des basses et des hautes fréquences en même temps.
- La représentation permet de représenter des micro-surfaces anisotropes.
- La représentation est compatible avec de nombreuses fonctions procédurales.
- Nous montrons que la représentation est compatible avec l'animation de la géométrie du modèle.

A voir

- L'article *LEADR Mapping* [DHI+13], son supplemental material et ses deux vidéos.

Chapitre 9

Représentation et évaluation du matériau multi-échelles

Dans le chapitre précédent, nous avons choisi une représentation pour la surface multi-échelles modélisée par une height map. Mais ces informations ne sont pas suffisantes pour calculer l'apparence de la surface qui ne dépend pas seulement de sa géométrie mais aussi du matériau en lequel elle est faite. L'objet de ce chapitre est de choisir une classe de matériaux pour la surface et de savoir comment la représenter aussi de manière multi-échelles.

Dans la section 9.1, nous rappelons comment le matériau de base de la micro-surface intervient dans le modèle de BRDF à micro-facettes que nous utilisons.

Dans la section 9.2, nous choisissons une BRDF à micro-facettes pour le matériau de la micro-surface et nous proposons une manière de le représenter. Nous montrons que l'évaluation du matériau multi-échelles, qui dépend du point de vue et des corrélations possibles avec les attributs de la surface, peut toujours se formuler comme un problème de filtrage de *color mapping*.

Dans le but de résoudre ce problème de filtrage de color mapping, dans la section 9.3, nous commençons par revoir les approches existantes pour approximer le calcul du filtrage des color maps appliquées sur des textures. Nous dérivons ensuite l'expression exacte de ce filtrage et nous proposons une manière simple de l'évaluer.

Enfin, dans la section 9.4, nous appliquons la même méthodologie au filtrage des matériaux des surfaces en tenant compte des corrélations possibles avec les attributs géométriques de la surface et, en conséquence, de la visibilité.

9.1 Le matériau dans les modèles à micro-facettes

9.1.1 Micro-surface à matériau constant

Dans le chapitre 5, nous avons vu que la macro-BRDF du modèle décentré est donnée par :

$$\rho(\omega_o, \omega_i) = \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2(\omega_o, \omega_i, p_m) dp_m, \quad (9.1)$$

où $G_2(\omega_o, \omega_i, p_m)$ et $\rho_{\mathcal{M}}(\omega_i, \omega_o, p_m)$ sont des fonctions spatiales définies sur la micro-surface \mathcal{M} . Pour passer à une intégrale statistique, les modèles de BRDFs à micro-facettes supposent généralement que le matériau de la micro-surface est constant sur \mathcal{M} , c'est-à-dire que la fonction $\rho_{\mathcal{M}}(\omega_o, \omega_i, p_m)$ ne dépend que de l'orientation de la normale $\omega_m(p_m)$ indépendamment de la localisation du point p_m , ou encore, de manière moins restrictive, que le matériau associé à la normale ω_m peut varier en fonction de p_m , mais d'une manière indépendante

des autres paramètres, ce qui s'écrit :

$$\begin{aligned}\rho(\omega_o, \omega_i) &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2(\omega_o, \omega_i, p_m) dp_m \\ &\approx \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \left(\frac{\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, p_m) dp_m}{\int_{\mathcal{M}} dp_m} \right) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2(\omega_o, \omega_i, p_m) dp_m,\end{aligned}$$

où $(\int_{\mathcal{M}} \rho_{\mathcal{M}}(\omega_o, \omega_i, p_m) dp_m)$ est le matériau moyen de la micro-surface. Cette manière d'écrire signifie que si le matériau n'était pas constant sur la micro-surface, alors il pourrait d'abord être moyenné globalement avant d'être composé avec les autres termes qui entrent en jeu dans l'expression de la macro-BRDF. Lorsque $\rho_{\mathcal{M}}$ n'est pas corrélé aux autres termes, cette manière d'écrire les choses est exacte. En revanche, sur certains types de surface, il existe des corrélations entre $\rho_{\mathcal{M}}$ et la visibilité G_2 ou encore avec les normales qui interviennent dans les produits scalaires. Dans ce cas, il n'est plus possible de faire une telle approximation.

Problème à résoudre. Pour pouvoir concrètement faire un calcul moins approximatif, il nous faut d'abord commencer par trouver une manière de représenter effectivement la micro-BRDF $\rho_{\mathcal{M}}(\omega_o, \omega_i, p_m)$ qui intervient dans le calcul. C'est l'objet de la section suivante.

9.2 Représentation du matériau multi-échelles

9.2.1 Représentation du matériaux de base

Dans cette section, nous définissons la manière de représenter le matériau de base de la surface, celui qui est donné à l'échelle la plus fine, en pratique stocké dans les textures à la plus haute résolution. Nous choisissons de représenter le matériau de base de la surface avec une BRDF à micro-facettes, donnée par un albédo de surface a et une distribution de micro-facettes P^{22} , généralement donnée par un ou plusieurs paramètres de rugosité et une classe de distribution choisie à l'avance. Nous choisissons d'utiliser la même représentation que pour la surface multi-échelles : une distribution de Beckmann représentée par sa moyenne et sa matrice de covariance.

Motivation. Cette manière de représenter les matériaux des surfaces est aujourd'hui la plus courante, aussi bien dans les communautés du rendu en temps-réel que dans le rendu offline [MHH⁺12].

Représentation mathématique. Nous représentons la rugosité du matériau par sa distribution de pentes P^{22} , que nous représentons de la même manière que la rugosité de la surface multi-échelles : avec une distribution gaussienne. Il suffit donc de représenter ses moments d'ordre 2 (la rugosité de base du matériau étant centrée, ses moments d'ordre 1 valent 0 et n'ont pas besoin d'être stockés). L'albédo de la surface est représentable par une simple valeur scalaire a .

Incorporation de la BRDF multi-échelles. Dans le chapitre 5, nous avons vu que lorsque la BRDF des micro-facettes est elle-même une BRDF à micro-facettes, alors la BRDF globale est une BRDF à micro-facettes dont la distribution des pentes est la convolution entre les deux distributions de pentes. En pratique, nous supposons que les BRDFs des micro-facettes sont aussi des BRDFs à micro-facettes à distribution gaussienne. Il s'agit donc de convoluer deux distributions gaussiennes. Cela se fait simplement en additionnant leurs moments.

Représentation multi-fréquences. En pratique, l'albédo de la surface varie en fonction de la fréquence de la lumière incidente et nous pouvons stocker plusieurs albédos de surface (typiquement représentés par des valeurs RGB). Sans perdre en généralité, nous ne décrivons dans ce chapitre que le traitement d'un seul albédo et d'une seule rugosité à la fois.

9.2.2 Se ramener à une représentation linéairement combinable et séparable

Comme expliqué dans le chapitre 7, une propriété essentielle de la représentation est que ses différents paramètres soient linéairement combinables et séparables. De cette manière, chacun des paramètres peut être stocké dans une texture interpolable et filtrable indépendamment des autres.

Albédo et rugosité indépendants. L'albédo a de la surface et les moments de P^{22} sont déjà linéairement combinables. Lorsqu'ils sont indépendants (non corrélés), ils peuvent être stockés séparément, tels quels, dans des textures filtrables.

Albédo et rugosité dépendants. Lorsque a et P^{22} sont corrélés, ils ne peuvent pas être filtrés séparément. Une manière de linéariser la représentation est de la décomposer dans une base dont les coefficients sont linéairement combinables. Par exemple, si tous les couples (a, P^{22}) existants sont représentables comme une combinaison linéaire de n couples de base :

$$(a, P^{22}) = \sum_{i=1}^n \lambda_i (a_i, P_i^{22}),$$

alors, par linéarité de a et P^{22} , l'évaluation de la BRDF avec les paramètres (a, P^{22}) est équivalente à l'évaluation de la BRDF pour les n matériaux de base et combinés avec les poids λ_i . Dans ce cas, plutôt que de stocker directement a et P^{22} , nous stockons les n coefficients λ_i , qui sont linéairement combinables et séparables.

Expression du problème du filtrage avec des paramètres de matériau génériques. L'idée principale de cette section est qu'il est important de se ramener à une représentation dont chacun des paramètres peut être filtré séparément des autres. C'est toujours le cas de a et des paramètres de P^{22} lorsqu'ils sont indépendants. Nous avons donné l'idée générale du processus de linéarisation lorsqu'ils le sont (nous ne la détaillerons pas plus dans cette thèse).

En conclusion, à partir de maintenant et dans la suite de ce chapitre, nous partirons toujours du principe que les paramètres à filtrer sont représentables d'une manière linéairement combinable et séparable. De fait, par la suite, nous nous poserons uniquement la question de savoir comment filtrer un attribut générique C , que nous appellerons par commodité la "couleur" ou qui sera parfois une fonction que nous appellerons une "color map", mais qui peut potentiellement représenter tout ce qui est linéairement combinable et séparable (albédo a , moments de la rugosité P^{22} , paramètres de poids λ , etc.).

En utilisant ces paramètres, l'équation (9.1) qui donne le matériau moyen se réécrit :

$$\bar{\rho}(C_1, \dots, C_n) = \frac{|\omega_o \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\mathcal{M}} \rho(\omega_o, \omega_i, C_1(p_m), \dots, C_n(p_m)) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1(\omega_o, p_m) G_1(\omega_i, p_m) dp_m.$$

Par ailleurs, puisque les paramètres sont linéairement combinables, la moyenne du matériau paramétré égale le matériau des paramètres moyens :

$$\bar{\rho}(C_1, \dots, C_n) = \rho(\bar{C}_1, \dots, \bar{C}_n).$$

En conclusion, pour calculer le matériau moyen $\bar{\rho}(C_1, \dots, C_n)$, il faut être capable d'évaluer les paramètres moyens des matériaux, donnés par :

$$\bar{C} = \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1(\omega_o, p_m) G_1(\omega_i, p_m) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1(\omega_o, p_m) G_1(\omega_i, p_m) dp_m}. \quad (9.2)$$

Problème à résoudre. Lorsque les attributs C ne sont corrélés ni avec les normales ni avec la visibilité, ils peuvent être directement stockés dans une texture filtrable. En cas de corrélation, leur filtrage doit être fait de manière dépendante du point de vue. Pour gérer ces corrélations, nous montrons qu'il est toujours possible de reformuler le problème du filtrage en passant par une fonction intermédiaire : les *color maps*.

9.3 Filtrage des color maps appliquées sur des textures

Nous verrons dans les sections suivantes comment filtrer les color maps lorsque celles-ci sont appliquées sur des attributs corrélés avec les normales ou la visibilité. Mais dans un premier temps, nous nous intéressons au problème du filtrage des color maps appliquées sur un attribut générique f de la surface qui n'est corrélé ni aux normales, ni à la visibilité, et peut donc être, dans l'équation (9.2), sorti de l'intégrale :

$$\begin{aligned} \bar{C} &= \frac{\int_{\mathcal{M}} C(f(p_m)) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1(\omega_o, p_m) G_1(\omega_i, p_m) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1(\omega_o, p_m) G_1(\omega_i, p_m) dp_m} \\ &= \frac{\int_{\mathcal{M}} C(f(p_m)) dp_m}{\int_{\mathcal{M}} dp_m} \frac{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1(\omega_o, p_m) G_1(\omega_i, p_m) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1(\omega_o, p_m) G_1(\omega_i, p_m) dp_m} \\ &= \frac{\int_{\mathcal{M}} C(f(p_m)) dp_m}{\int_{\mathcal{M}} dp_m}. \end{aligned}$$

Dans cette section, nous cherchons donc à résoudre l'équation :

$$\frac{\int_{\mathcal{M}} C(f(p_m)) dp_m}{\int_{\mathcal{M}} dp_m} = ?$$

où C est une color map et f est une fonction spatiale que nous appellerons “fonction de bruit” pour dénoter le fait qu'elle n'est pas corrélée à des propriétés de la surface telles que ses normales ou sa visibilité. Nous appellerons aussi la fonction composée $C(f)$ une “texture procédurale” car c'est souvent de cette manière que les textures procédurales sont créées en pratique. Dans cette section, nous développons une méthodologie générale pour résoudre ce problème de filtrage de color mapping. Dans la section suivante, nous appliquerons la même méthodologie lorsque la color map est une fonction appliquée sur les hauteurs et les normales de la micro-surface et dont le filtrage devient dépendant de la visibilité.

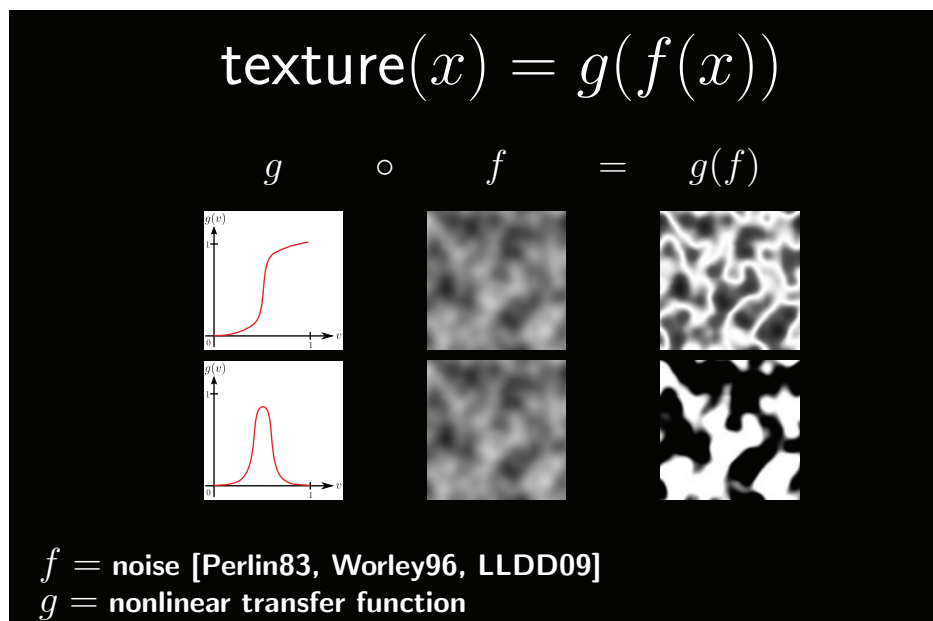


FIGURE 9.1 – Appliquer une opération non linéaire g sur une fonction de bruit f est une étape classique pour créer des textures procédurales.

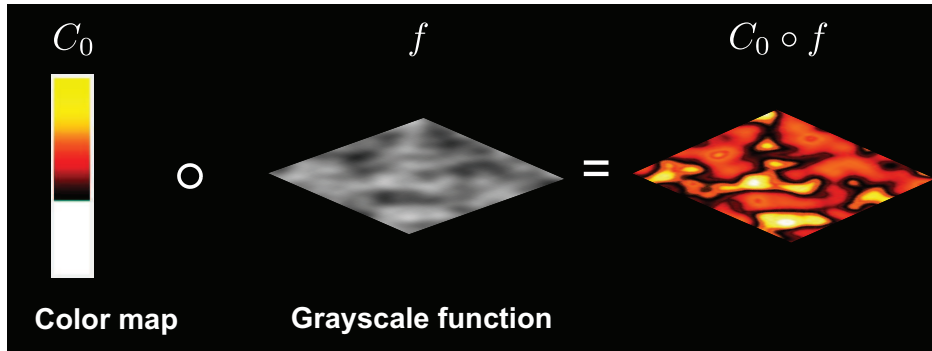


FIGURE 9.2 – Dans le contexte des textures procédurales, nous appelons “color map” la fonction C résultante finale de l’ensemble des opérations non linéaires appliquées sur une “fonction de bruit” f .

9.3.1 Rappels sur le filtrage des textures procédurales

Notre motivation première pour explorer ce problème de filtrage de color maps est qu’il est lié au filtrage des surfaces complexes. Nous ouvrons une parenthèse pour décrire ce problème plus en détail, car même s’il ne constitue qu’une étape intermédiaire dans les dérivations de ce chapitre, c’est un problème intéressant en soi et non résolu dans la littérature sur les représentations procédurales. En effet, les représentations procédurales sont celles dont les données ne sont pas stockées en mémoire mais évaluées à la volée, typiquement dans des shaders. Elles sont commodes pour ajouter des détails haute-résolution aux surfaces en utilisant très peu de mémoire [Per85, Pea85].

De la même manière que les textures standard, les textures procédurales doivent aussi être filtrées pour obtenir la bonne apparence et ne pas introduire d’aliasing. Comme elles ne sont pas stockées explicitement en mémoire, il n’est pas possible d’utiliser une représentation pré-filtrée, comme le MIPmapping, et l’opération de filtrage doit se faire à la volée, au moment de l’évaluation de la texture procédurale.

En général, les textures procédurales utilisent des fonctions de bruit f comme primitives de base. Ces fonctions de bruit sont souvent, par construction, facilement filtrables [LLC⁺10, LD11], c’est-à-dire que la valeur de $\int f(x)dx$ peut être évaluée directement depuis les propriétés du processus générateur de f .

Matériaux appliqués sur une fonction de bruit multi-échelles. Mais les fonctions de bruit sont rarement utilisées directement comme textures procédurales. Elles sont souvent passées dans différents types de filtres qui sculptent l’apparence de la texture par-dessus le bruit, comme illustré dans la figure 9.1. Les *color maps* sont une manière commode de représenter la somme des opérations que subissent les valeurs de la fonction de bruit [Mus02], voir figure 9.2. En général, la texture s’exprime comme la fonction composée $C(f(x))$ où C est la color map appliquée sur la fonction de bruit.

Le filtrage des textures procédurales sur lesquelles on applique une color map est donné par l’intégrale de la fonction composée $\int_{\mathcal{M}} C(f(p_m)) dp_m$. Puisque les color maps sont souvent des fonctions non linéaires, dans le cas général, on ne peut pas simplement appliquer la color map sur le bruit filtré : $\int C(f(x))dx \neq C(\int f(x) dx)$. Des revues récentes [LLC⁺10, BN12a], décrivent le problème du filtrage des color maps comme un problème non résolu.

Transition globale (*global fading*). Les méthodes de simplification de shaders [HSS98, OKS03, Pel05] mélangent progressivement les valeurs de la texture procédurale avec sa valeur globale moyenne. L’interpolation entre la couleur de la texture et la couleur moyenne commence lorsque les fréquences de la texture sont supérieures à la fréquence d’échantillonnage des pixels, de manière à limiter l’aliasing. Le problème de cette technique est qu’elle supprime toutes les fréquences en même temps de la même manière. Mais en théorie, les hautes fréquences devraient être supprimées progressivement, selon la taille du filtre. Cette manière de faire introduit donc un compromis artificiel entre la réduction de l’aliasing (atténuer beaucoup la texture)

et la préservation des détails : réduire trop tôt la pondération de la texture produit un effet de flou visible, la réduire trop tard introduit de l'aliasing. Puisque toutes les fréquences sont traitées de la même manière en même temps, il n'existe pas de solution idéale entre trop tôt et trop tard.

Intégration numérique. Worley utilise une méthode d'intégration numérique pour filtrer la color map [Wor02]. C'est une approche "brute force". Elle peut par exemple être réalisée de manière adaptative avec des heuristiques sur les spectres des fonctions composées [BMW06]. Mais, comme nous l'avons expliqué dans le chapitre d'introduction, les méthodes adaptatives ne sont pas scalables : la complexité du calcul n'est pas indépendante de la complexité des données.

Filtrage de la color map. Certaines techniques approximent le filtrage de la fonction composée $C(f)$ par une méthode de filtrage appliquée sur la fonction C . En pratique, la color map est représentée avec des niveaux de MIPmap [Wil83] et le niveau de MIPmap est choisi en fonction du gradient de la fonction de bruit [RTB⁺92, HCK⁺99]. Cette technique approxime localement la fonction de bruit avec un développement limité d'ordre 1. Elle ne fonctionne donc que lorsque la taille des pixels est petite relativement aux fréquences de la texture. Lagae. et al. utilisent la variance des valeurs de f sous-pixel pour paramétrer un filtre-boîte appliqué sur la color map [LLC⁺10]. Cette méthode est très proche de la méthode exacte que nous proposons. Nous expliquons pourquoi dans la section suivante.

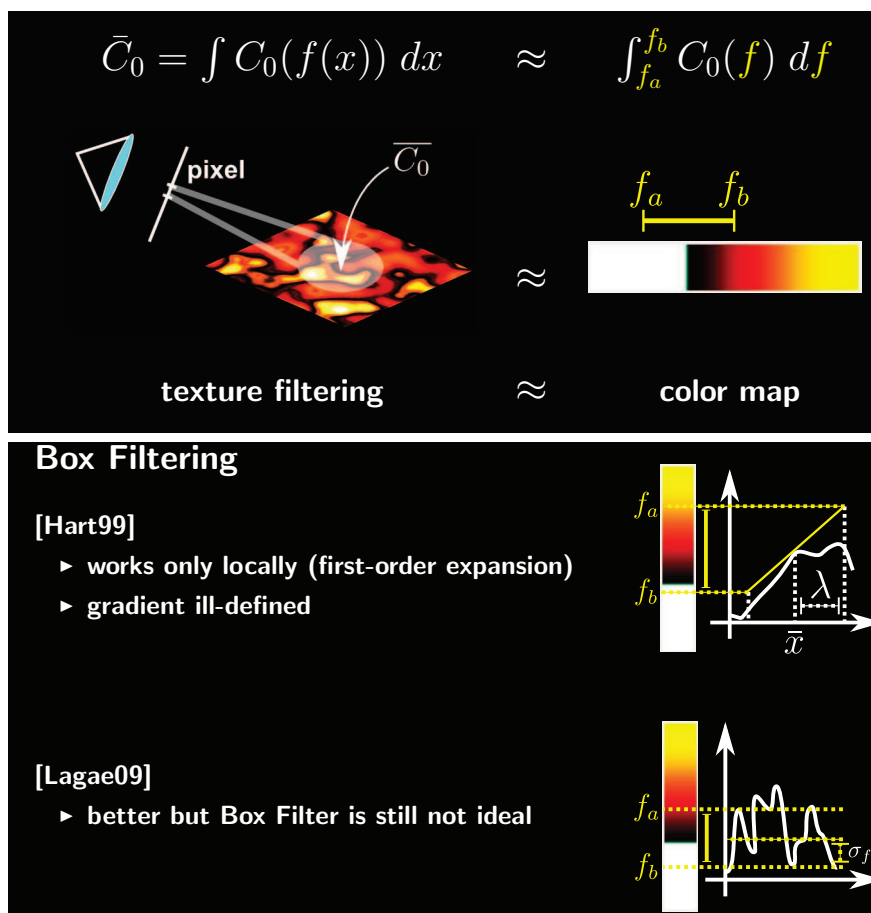


FIGURE 9.3 – Certaines approches précédentes filtrent la texture procédurale $C(f)$ en paramétrant un filtre boîte appliqué sur la color map C avec différentes heuristiques calculées à partir de la fonction f .

9.3.2 Dérivation exacte de l'équation du filtrage des color maps

Remplacer la formulation spatiale par une formulation statistique. Dans le chapitre 4, nous avons expliqué que la théorie des micro-facettes est une formulation statistique des interactions entre la lumière et la micro-surface. Pour exprimer cette idée mathématiquement, nous avons transformé les intégrales spatiales qui décrivent le transport de la lumière en des intégrales statistiques. Formellement, nous avons changé l'espace d'intégration en remplaçant l'espace de la micro-surface \mathcal{M} par l'espace de ses propriétés statistiques, donné par l'espace sphérique Ω dans lequel vivent les normales de la micro-surface. Pour opérer ce changement d'espace, nous avons utilisé une mesure qui garantit la conservation des aires lorsqu'on change le domaine d'intégration : pour les micro-facettes, c'est la distribution de normales $D(\omega_m)$.

Nous faisons de même avec les textures procédurales, à la différence que la quantité pertinente n'est plus les normales de la micro-surface, mais les valeurs de la fonction de bruit f . Pour exprimer le changement d'espace, nous avons besoin de caractériser l'espace dans lequel vivent les valeurs de la fonction f et la distribution statistique des valeurs de f dans cet espace. En général, l'espace des valeurs f de bruit est simplement l'ensemble des nombres réels $f \in]-\infty, +\infty[$ (mais ce pourrait être n'importe quoi d'autre, éventuellement avec plusieurs dimensions) et nous notons D_f la distribution associée aux valeurs $f(p_m)$ sur le domaine spatial \mathcal{M} , donnée par :

$$D_f(f') = \int_{\mathcal{M}} \delta_{f'}(f(p_m)) dp_m.$$

Maintenant que nous avons l'espace et la mesure, nous pouvons reformuler l'équation du filtrage de la texture procédurale dans cet espace-là :

$$\begin{aligned} \frac{\int_{\mathcal{M}} C(f(p_m)) dp_m}{\int_{\mathcal{M}} dp_m} &= \frac{\int_{-\infty}^{+\infty} C(f) D_f(f) df}{\int_{-\infty}^{+\infty} D_f(f) df} \\ &= \int_{-\infty}^{+\infty} C(f) \frac{D_f(f)}{\int_{-\infty}^{+\infty} D_f(f) df} df. \end{aligned} \quad (9.3)$$

Ce résultat montre que la valeur de la texture procédurale filtrée est le produit scalaire¹ entre la color map C et la distribution normalisée $\frac{D_f(f)}{\int_{-\infty}^{+\infty} D_f(f) df}$ des valeurs de bruit. Cette idée est illustrée dans la figure 9.4.

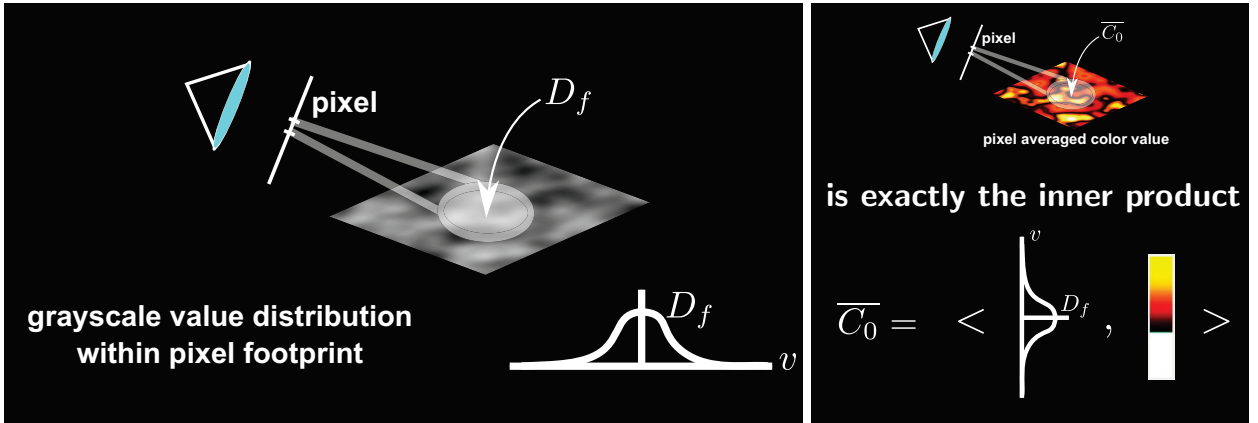


FIGURE 9.4 – La valeur $\int C(f)$ de la texture procédurale filtrée est le produit scalaire entre la distribution des valeurs de bruit D_f et la color map C .

1. L'intégrale $\int a(x)b(x)dx$ d'un produit est un produit scalaire.

9.3.3 Représentation et calcul

Problème. Pour utiliser le résultat de l'équation (9.3) en pratique, dans le but de rendre des textures procédurales filtrées, il reste deux problèmes à résoudre : la représentation de D_f et le calcul du produit scalaire $\int_{-\infty}^{+\infty} C(f) D_f(f) df$. Dans l'état d'esprit du cahier des charges présenté au chapitre 7, nous souhaitons trouver une représentation simple pour D_f et une manière efficace de faire l'évaluation du produit scalaire.

Notre méthode. Nous proposons de représenter D_f par une distribution gaussienne, donnée par ses moments $\mathbb{E}[f]$ et $\mathbb{E}[f^2]$. C'est une représentation simple, légère et surtout souvent pertinente, car la plupart des fonctions de bruit produisent effectivement des statistiques gaussiennes, ce qui garantit que notre représentation tend vers la distribution exacte lorsque le voisinage \mathcal{M} dans laquelle elle est calculée est suffisamment grand. De plus, certaines fonctions de bruit donnent par construction directement accès aux valeurs de ces deux moments.

Pour évaluer le produit scalaire, nous utilisons une color map pré-convoluée avec des noyaux gaussiens de différentes tailles. Nous représentons la color map pré-filtrée de cette manière dans une table 2D :

$$C(\mathbb{E}[f], \sigma_f) = \sqrt{\mathbb{E}[f^2] - \mathbb{E}^2[f]},$$

qui donne la valeur du produit scalaire entre la color map et la distribution gaussienne $\mathcal{N}(\mathbb{E}[f], \sigma_f^2)$ paramétrée par sa moyenne $\mathbb{E}[f]$ et son écart-type σ_f . Cet algorithme est illustré en figure 9.5.

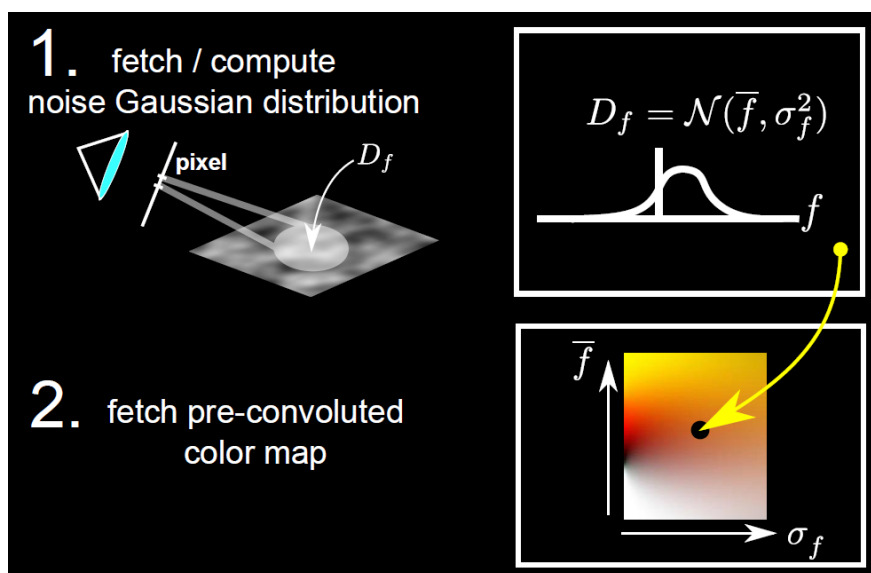


FIGURE 9.5 – Notre algorithme de rendu multi-échelles de textures procédurales color mappées. Nous utilisons une color map pré-convoluée avec un filtre gaussien que nous paramétrons avec les statistiques gaussiennes de la fonction de bruit.

Il est à noter que le travail concurrent de Hadwiger et al. [HSB⁺12] se base sur la même idée mais utilise des représentations plus complexes pour les distributions, qui n'ont pas les propriétés que nous recherchons (calculable à la volée, interpolable, etc.). Ce travail a été rendu public au moment où le nôtre [HNPN13a] était en cours de relecture pour I3D.

9.3.4 Résultats

La figure 9.6 illustre des résultats obtenus avec notre technique de filtrage de textures procédurales.

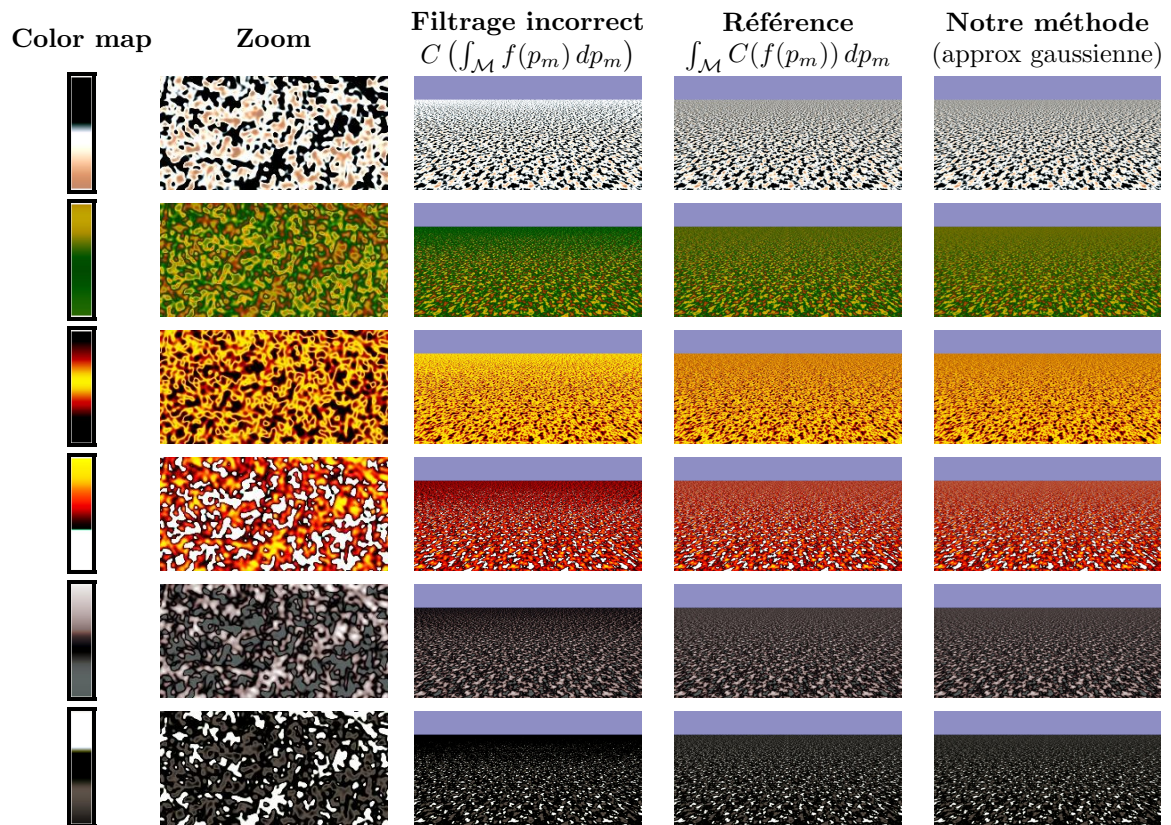


FIGURE 9.6 – Filtrage de textures qui sont la composition d'un bruit gaussien avec une color map.

9.4 Filtrage des color maps appliquées sur des surfaces

9.4.1 Idées générales

Nous avons vu que lorsque les paramètres de matériau ne sont pas corrélés avec les normales ou la fonction d'ombrage et de masquage de la micro-surface, alors ils peuvent être directement filtrés. En revanche, lorsqu'ils présentent des corrélations, le problème est plus compliqué. Le but de cette section est de montrer comment représenter ces corrélations et les introduire effectivement dans le calcul. Nous montrons dans cette section que représenter ces corrélations équivaut à décrire des color maps appliquées sur des descripteurs géométriques de la micro-surface : les normales et les hauteurs.

Motivation pour la formulation avec des color maps. Nous allons voir que cette formulation à base de color maps appliquées sur les normales et les hauteurs de la micro-surface émerge toute seule des dérivations mathématiques. Mais il est à noter que cette formulation est par ailleurs très commode. En effet, corrélérer les matériaux de la surface avec des descripteurs géométriques est très naturel et fait souvent partie des paradigmes classiques de création de contenu. A titre d'exemple, la figure 9.7 illustre une composition de texture de terrain, où les paramètres de la composition sont des fonctions des hauteurs et des pentes du terrain (nous rappelons que les pentes et les normales sont équivalentes sur une height map).

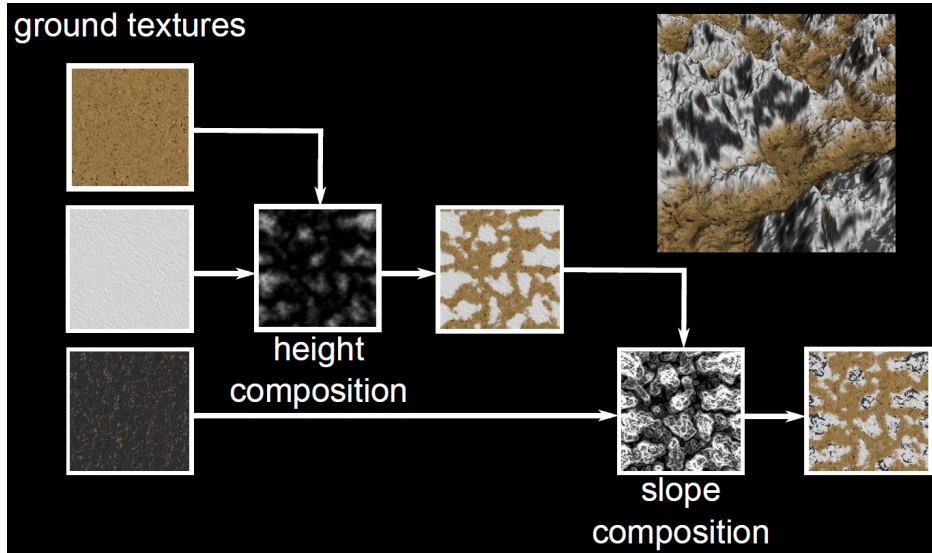


FIGURE 9.7 – Corréler les couleurs à des descripteurs géométriques tels que les hauteurs et les pentes d'une surface est une manière naturelle de créer des textures appliquées sur ces surfaces.

Emergence de la corrélation avec les hauteurs et les normales. Nous avons vu dans le chapitre 4 que les fonctions de masquage et d'ombrage s'expriment sous la forme :

$$G_1(\omega_o, p_m) = \chi^+(\omega_o \cdot \omega_m) G_1^{\text{dist}}(\omega_o, p_m),$$

$$G_1(\omega_i, p_m) = \chi^+(\omega_i \cdot \omega_m) G_1^{\text{dist}}(\omega_i, p_m),$$

où $G_1^{\text{dist}}(\omega_o, p_m)$ et $G_1^{\text{dist}}(\omega_i, p_m)$ sont respectivement les fonctions de masquage et d'ombrage distantes. En utilisant ce développement, l'équation (9.2) peut se récrire :

$$\bar{C} = \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1^{\text{dist}}(\omega_o, p_m) G_1^{\text{dist}}(\omega_i, p_m) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_1^{\text{dist}}(\omega_o, p_m) G_1^{\text{dist}}(\omega_i, p_m) dp_m},$$

où nous avons omis les fonctions de Heaviside qui sont redondantes avec les produits scalaires clampés. Nous utilisons maintenant deux propriétés du modèle de Smith. D'une part, grâce au chapitre 6, nous savons intégrer très précisément la fonction jointe de masquage et d'ombrage distante $G_2^{\text{dist}}(\omega_o, \omega_i, p_m) = G_1^{\text{dist}}(\omega_o, p_m) G_1^{\text{dist}}(\omega_i, p_m)$. A partir de maintenant, nous traiterons toujours l'ombrage et le masquage sous cette forme :

$$\bar{C} = \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m}.$$

D'autre part, nous avons vu dans le chapitre 4 que la propriété fondamentale du modèle de Smith est que la **fonction d'ombrage et de masquage distante** $G_2^{\text{dist}}(\omega_o, \omega_i, p_m)$ est indépendante des **normales** ω_m . Nous réécrivons l'équation en couleurs pour montrer les corrélations mises en jeu :

$$\bar{C} = \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m}. \quad (9.4)$$

Les termes en rouge peuvent être intégrés séparément des termes en bleu. C'est une propriété intrinsèque au modèle de Smith. Enfin, dans le modèle de Smith, la fonction de masquage et d'ombrage distante statistique est une fonction qui ne dépend que des hauteurs de la micro-surface (voir annexe A.2). Le seul terme qui ne soit pas en couleur est le paramètre du matériau C . Il y a donc deux manières évidentes de traiter les corrélations avec C et le reste : celles qui viennent de **fonction d'ombrage et de masquage distante**, donc des **hauteurs**, et celles qui viennent des **normales**.

Dans la section 9.4.2, nous montrons que les corrélations avec le masquage et l'ombrage peuvent être représentées à travers une color map statistique C_h appliquée sur les hauteurs h , comme illustré dans la figure 9.8. Dans la section 9.4.3 ; nous montrons que les corrélations avec les normales peuvent être représentées à travers une color map statistique C_{ω_m} appliquée sur les normales ω_m , comme illustré dans la figure 9.9.

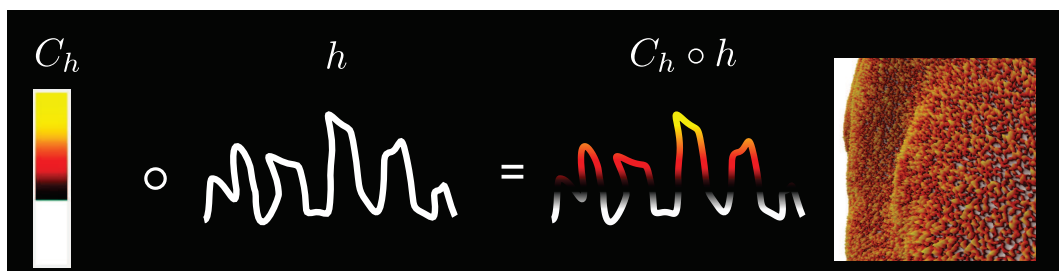


FIGURE 9.8 – Nous représentons les paramètres des matériaux corrélés avec les hauteurs de la micro-surface sous la forme d'une color map $C_h(h)$ appliquée sur les hauteurs de la micro-surface.

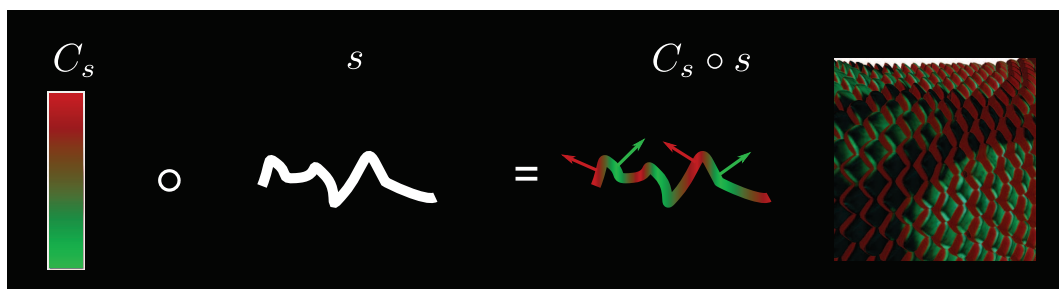


FIGURE 9.9 – Nous représentons les paramètres des matériaux corrélés avec les normales de la micro-surface sous la forme d'une color map $C(\omega_m)$ appliquée sur les normales (ou les pentes) de la micro-surface.

Algorithme générique. Nous avons vu en section 9.3 comment filtrer des textures obtenues en appliquant des color maps sur une fonction de bruit. Maintenant que nous savons résoudre ce problème, nous pouvons nous inspirer de sa solution pour résoudre le problème du filtrage des color maps appliquées sur des surfaces. Pour filtrer une color map appliquée sur des hauteurs ou des normales, nous appliquons la même idée que dans la section 9.3 : nous reformulons l'intégrale spatiale du filtrage en une intégrale statistique exprimée dans l'espace où est définie la color map. En revanche, contrairement au filtrage des textures présenté dans la section précédente, la visibilité intervient sur les surfaces, puisque la fonction G_2^{dist} et les produits scalaires clampés $\langle \omega_o, \omega_m, \cdot \rangle$ et $\langle \omega_i, \omega_m, \cdot \rangle$ font aussi partie des intégrales de filtrage. Nous allons voir dans les sections suivantes que le filtrage de la color map est le produit scalaire, non avec les distributions de hauteurs et de normales de la micro-surface, mais avec les distributions *visibles* de hauteurs et de normales. Ces distributions sont illustrées en figure 9.10 et sont utilisées dans l'algorithme de filtrage des surfaces color mappées présenté en figure 9.11. Pour appliquer cette idée générale avec des hauteurs et/ou des pentes color mappées en pratique, il faut savoir comment représenter les color maps ainsi que les distributions de hauteurs et de normales visibles. C'est l'objet des deux sections suivantes.

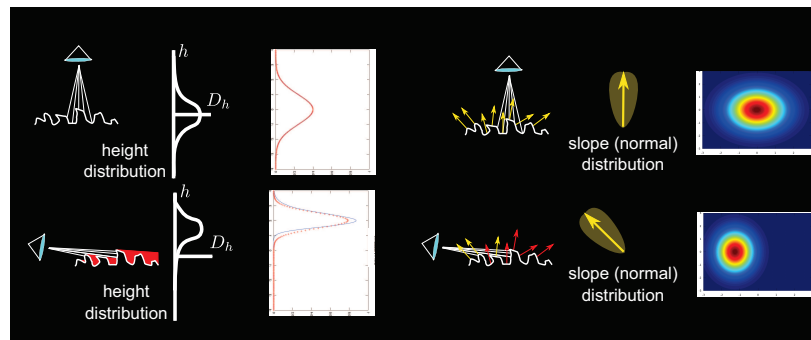


FIGURE 9.10 – Les statistiques visibles d'une surface sont différentes des statistiques intrinsèques de la surface, en particulier aux angles rasants.

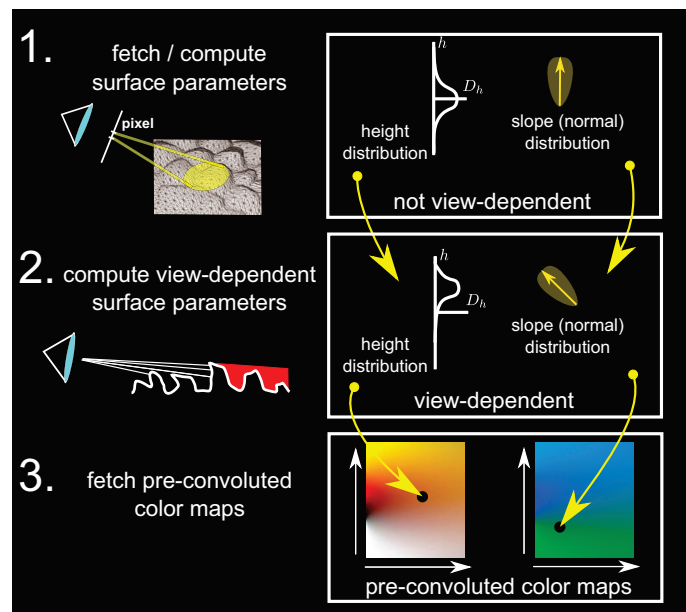


FIGURE 9.11 – Notre algorithme de rendu multi-échelles de surfaces color mappées. Nous calculons les statistiques de la micro-surface visible et les utilisons pour accéder à une color map pré-convoluée.

9.4.2 Filtrage des color maps appliquées sur les hauteurs

Matériaux corrélés avec les hauteurs. Dans cette section, nous cherchons à calculer la valeur \bar{C} du paramètre du matériau donné par l'équation (9.4) lorsque celui-ci est corrélé uniquement avec les hauteurs de la micro-surface :

$$\begin{aligned}
\bar{C} &= \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m} \\
&= \frac{\int_{\mathcal{M}} C(p_m) G_1^{\text{dist}}(\omega_o, p_m) G_1^{\text{dist}}(\omega_i, p_m) dp_m \int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle dp_m}{\int_{\mathcal{M}} G_1^{\text{dist}}(\omega_o, p_m) G_1^{\text{dist}}(\omega_i, p_m) dp_m \int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle dp_m} \\
&= \frac{\int_{\mathcal{M}} C(p_m) G_1^{\text{dist}}(\omega_o, p_m) G_1^{\text{dist}}(\omega_i, p_m) dp_m}{\int_{\mathcal{M}} G_1^{\text{dist}}(\omega_o, p_m) G_1^{\text{dist}}(\omega_i, p_m) dp_m}. \tag{9.5}
\end{aligned}$$

L'idée est de reformuler cette intégrale spatiale comme une intégrale statistique, cette fois-ci dans l'espace des hauteurs de la surface. En effet, dans le modèle de Smith, la fonction d'ombrage et de masquage statistique ne dépend que des hauteurs de la micro-surface. Cette fonction statistique définie dans l'espace des hauteurs s'écrit $G_2^{\text{dist}}(\omega_o, \omega_i, h)$. En partant de cette observation, il est naturel de traiter les corrélations entre le matériau C et la fonction G_2^{dist} en exprimant l'intégrale dans l'espace des hauteurs.

Color map statistique des hauteurs. Pour exprimer l'intégrale statistique en espace des hauteurs, nous devons exprimer la manière dont les paramètres de matériau C , se répartissent par hauteur et par normale. Cette répartition est donnée par la color map statistique des hauteurs et des normales, donnée par :

$$C_h(h') = \frac{\int_{\mathcal{M}} \delta_{h'}(h(p_m)) C(p_m) dp_m}{\int_{\mathcal{M}} \delta_{h'}(h(p_m)) dp_m}, \tag{9.6}$$

où la valeur $C_h(h)$ représente le paramètre de matériau moyen à la hauteur h . Il est à noter que cette color map est *locale* : elle dépend du voisinage \mathcal{M} , et *statistique* : cela ne signifie pas que toutes les hauteurs h de la micro-surface aient nécessairement la même couleur $C_h(h)$, mais que $C_h(h)$ est la couleur moyenne à la hauteur h .

Distribution statistique intrinsèque des hauteurs de la micro-surface. Les hauteurs h de la micro-surface sont des nombres réels $h \in]-\infty, +\infty[$ et leur distribution statistique intrinsèque sur la micro-surface est notée $P^1(h)$ (voir annexe A.2). Nous rappelons que dans le chapitre 8 nous avons fait le choix de représenter des surfaces à statistiques gaussiennes. Cela détermine la distribution $P^1(h)$: c'est la densité de probabilité d'une loi gaussienne $\mathcal{N}(\bar{h}, \sigma_h^2)$, dont les moments d'ordre 1 et 2 sont disponibles dans une texture MIPmappée ou calculables à la volée dans le cas d'une surface multi-échelles procédurale.

Intégration statistique de la color map appliquée sur les hauteurs. En utilisant les expressions statistiques dans l'espace des hauteurs de la color map $C(\omega_m, h)$ et de la fonction d'ombrage et de masquage $G_2^{\text{dist}}(\omega_o, \omega_i, h)$, l'intégrale spatiale de l'équation (9.5) se réécrit comme une intégrale statistique en espace des hauteurs :

$$\begin{aligned}
\bar{C} &= \frac{\int_{-\infty}^{+\infty} C_h(h) G_2^{\text{dist}}(\omega_o, \omega_i, h) P^1(h) dh}{\int_{-\infty}^{+\infty} G_2^{\text{dist}}(\omega_o, \omega_i, h) P^1(h) dh} \\
&= \int_{-\infty}^{+\infty} C_h(h) \frac{G_2^{\text{dist}}(\omega_o, \omega_i, h) P^1(h)}{\int_{-\infty}^{+\infty} G_2^{\text{dist}}(\omega_o, \omega_i, h) P^1(h) dh} dh \\
&= \int_{-\infty}^{+\infty} C_h(h) \frac{D_h(h)}{\int_{-\infty}^{+\infty} D_h(h) dh} dh.
\end{aligned}$$

Cela montre que, d'une manière similaire à l'équation 9.3 du filtrage des textures procédurales color mappées, le filtrage de la color map appliquée sur les hauteurs revient à calculer le produit scalaire entre la color map des hauteurs et une fonction D_h , que nous appelons la *distribution des hauteurs visibles* depuis ω_o et ω_i . C'est l'idée de l'algorithme illustré en figure 9.11.

Distribution des hauteurs visibles. La distribution des hauteurs visibles est définie par :

$$D_h(h) = G_2^{\text{dist}}(\omega_o, \omega_i, h) P^1(h).$$

Cette expression montre que la distribution des hauteurs visibles est simplement la distribution des hauteurs P^1 intrinsèque à la micro-surface pondérées par la fonction de masquage et d'ombrage G_2^{dist} : plus une hauteur a de chances d'être visible pour ω_o et ω_i , plus elle est importante. Le facteur de normalisation de cette distribution est donné par :

$$\int_{-\infty}^{+\infty} D_h(h) dh = \int_{-\infty}^{+\infty} G_2^{\text{dist}}(\omega_o, \omega_i, h) P^1(h) dh.$$

Dans le modèle de Smith, la fonction G_2^{dist} est donnée par la fonction Λ associée :

$$G_2^{\text{dist}}(\omega_o, \omega_i, h) = \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)},$$

et la distribution se réexprime :

$$D_h(h) = \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)} P^1(h).$$

Sous cette forme, la distribution D_h des hauteurs visibles est entièrement déterminée : nous savons évaluer chacun des termes de l'équation. Mais cette forme ne permet pas une évaluation analytique de l'équation du filtrage de la color map. Nous utilisons la même astuce qu'avec les distributions de bruit procédurales : nous l'approximons par une distribution gaussienne donnée par une moyenne $\bar{h}_{\omega_o\omega_i}$ et une variance $\sigma_{h\omega_o\omega_i}^2$. Nous utilisons ensuite ces deux paramètres pour accéder à une color map pré-convolué avec un filtre gaussien, comme expliqué dans l'algorithme de la figure 9.11.

Approximation gaussienne de la distribution des hauteurs visibles. Nous approximations D_h par une densité de probabilité d'une distribution gaussienne $\mathcal{N}(\bar{h}_{\omega_o\omega_i}, \sigma_{h\omega_o\omega_i}^2)$, où les paramètres $\bar{h}_{\omega_o\omega_i}$, $\sigma_{h\omega_o\omega_i}^2$ sont la moyenne et la variance de D_h . En pratique, nous approximations :

$$\begin{aligned} \bar{h}_{\omega_o\omega_i} &= \frac{\int_{-\infty}^{+\infty} h \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)} P^1(h) dh}{\int_{-\infty}^{+\infty} \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)} P^1(h) dh} \\ &\approx \bar{h} + \alpha_1 \sigma_h \log(\beta_1 [\Lambda(\omega_o) + \Lambda(\omega_i)] + 1.0), \end{aligned} \quad (9.7)$$

et :

$$\begin{aligned} \sigma_{h\omega_o\omega_i}^2 &= \frac{\int_{-\infty}^{+\infty} h^2 \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)} P^1(h) dh}{\int_{-\infty}^{+\infty} \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)} P^1(h) dh} - \mu_h^2 \\ &\approx \left[\frac{\sigma_h}{1.0 + \alpha_2 \log(\beta_2 (\Lambda(\omega_o) + \Lambda(\omega_i)) + 1.0)} \right]^2. \end{aligned} \quad (9.8)$$

Nous obtenons les paramètres α et β en utilisant un algorithme d'optimisation non linéaire, qui nous donne : $\alpha_\mu = 0.39, \beta_\mu = 4.75, \alpha_\sigma = 0.26$, et $\beta_\sigma = 1.13$.

Pour trouver l'approximation analytique de $\bar{h}_{\omega_o\omega_i}$, nous avons cherché une forme analytique qui respecte les propriétés suivantes :

- En incidence frontale ($\theta_i = \theta_o = 0$), alors $\Lambda(\omega_o) + \Lambda(\omega_i) = 0$ et la distribution visible est simplement la distribution de la surface $D_h = P^1$.
- Nous observons empiriquement que $\bar{h}_{\omega_o\omega_i}$ augmente linéairement avec σ_h , ce qui explique l'introduction d'un facteur multiplicatif.
- Nous observons empiriquement que $\bar{h}_{\omega_o\omega_i}$ augmente de manière monotone avec $[\Lambda(\omega_o) + \Lambda(\omega_i)]$ mais avec une dérivée qui décroît, ce qui explique le choix d'un terme en log.

Nous avons utilisé la même méthodologie pour $\sigma_{h\omega_o\omega_i}^2$.

La figure 9.12 compare les valeurs de D_h avec notre approximation sous cette forme avec ces paramètres α et β . A incidence frontale, nous obtenons la forme exacte de la distribution $D_h = P^1$. A incidence rasante, notre approximation n'est plus exacte, mais est précise : on observe le bon décalage (augmentation de la moyenne) et le bon resserrement de la distribution (diminution de la variance).

De plus, cette représentation est facile et très légère à calculer.

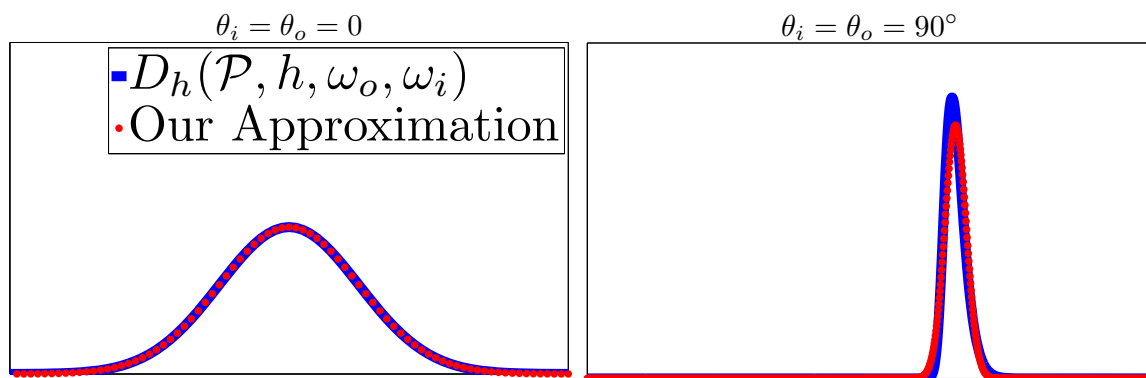


FIGURE 9.12 – Notre approximation analytique de la distribution $D_h(h)$ des hauteurs visibles en fonction de ω_o et ω_i . En incidence frontale, la distribution des hauteurs visibles est exactement la distribution des hauteurs de la micro-surface et notre approximation est exacte. En incidence rasante, notre approximation analytique n'est plus exacte mais capture le comportement global de la référence : décalage de la moyenne et resserrement de la distribution.

Corrélation entre le masquage et l'ombrage. Dans les dérivations précédentes, nous avons utilisé la forme azimuthalement décorrélée de la fonction de masquage et d'ombrage : les deux fonctions Λ sont additionnées. Il est important de noter que ce qui est développé dans cette section est parfaitement compatible avec le modèle d'ombrage et de masquage plus sophistiqué présenté au chapitre 6 : il suffirait de remplacer la forme additive par la forme plus complexe qui introduit la corrélation azimuthale dans l'ensemble des équations de cette section et en particulier dans les équations (9.7) et (9.8) qui servent à calculer les paramètres de D_h .

Utilisation d'une unique color map perturbée. En pratique, il est courant d'utiliser une seule et unique color map $C_h(h)$ pour l'ensemble de la surface, en particulier pour générer des données procédurales. Pour enrichir la représentation il est possible de "casser" légèrement la corrélation en perturbant le mapping sur les hauteurs en appliquant $C_h(h(p_m) + f(p_m))$, où $f(p_m)$ est une fonction de bruit. De cette manière, des points de hauteurs égales peuvent avoir des couleurs différentes, mais être quand même corrélés.

Dans ce cas, la distribution à utiliser dans l'équation du filtrage n'est pas celle des valeurs de $h(p_m)$, mais celle des valeurs de la somme $h(p_m) + f(p_m)$. Lorsque f est une fonction de bruit, nous avons vu dans la section précédente qu'elle est décorrélée de la fonction de hauteur h , et donc de la visibilité, et qu'elle s'approxime bien par une distribution gaussienne D_f de loi $\mathcal{N}(\bar{f}, \sigma_f^2)$ dont les paramètres sont connus ou calculables.

Puisque le bruit et les hauteurs ne sont pas corrélés, la distribution des valeurs de la somme $h(p_m) + f(p_m)$ est la convolution des distributions $D_h * D_f$. De plus, la convolution de deux gaussiennes est une gaussienne dont les paramètres sont les sommes des paramètres des deux distributions : $\mathcal{N}(\bar{h}_{\omega_o \omega_i} + \bar{f}, \sigma_{\omega_o \omega_i}^2 + \sigma_f^2)$.

Cette forme de color mapping perturbé est donc totalement compatible avec ce que nous avons décrit dans cette section : il suffit d'additionner les paramètres de D_h avec ceux de D_f pour obtenir les nouveaux paramètres qui servent à accéder à la color map pré-convoluée dans l'algorithme de la figure 9.11.

Résultats. Les figures 9.13 et 9.14 illustrent des résultats obtenus avec notre algorithme de filtrage des color maps appliquées sur les hauteurs. Pour générer ces résultats, nous avons utilisé une color map $C_h(h)$ appliquée sur les hauteurs, indépendante des normales, qui contient des valeurs RGB. Avec notre méthode, la couleur filtrée affichée est donnée par $\int_{-\infty}^{+\infty} C_h(h) D_h(h) dh$, où D_h est représentée avec notre approximation gaussienne et C_h est une fonction globale stockée sous la forme d'une table 2D pré-convoluée, indexée par la moyenne et l'écart-type de D_h .

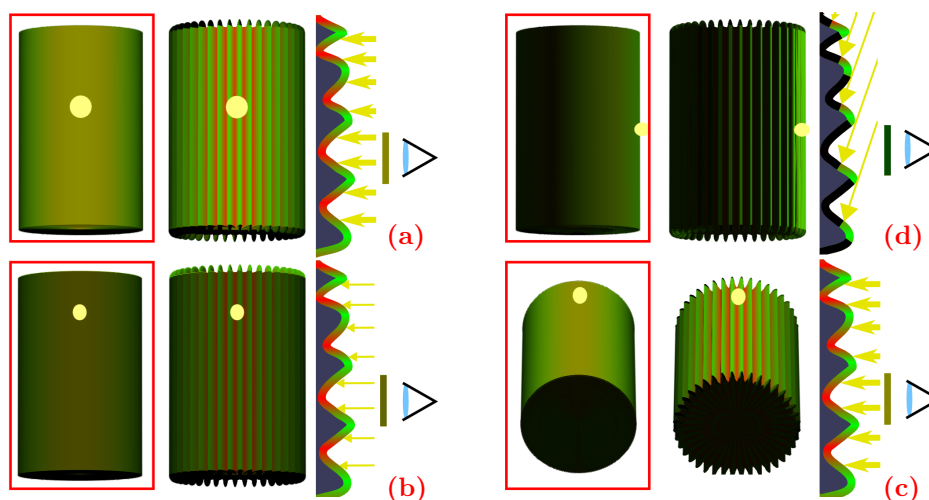


FIGURE 9.13 – Notre modèle de filtrage des color maps appliquées sur les hauteurs sur une surface anisotrope. Les images encadrées en rouge donnent la couleur moyenne visible (ω_o) et éclairée (ω_i) calculée avec notre approximation gaussienne. (a) Les directions d'observation ω_o et de l'éclairage ω_i sont frontales, il n'y a pas d'effets de visibilité. (b) La direction de l'éclairage ω_i est rasante mais dans une direction où la surface anisotrope n'est pas rugueuse, il n'y a pas d'effets de visibilité. (c) La direction d'observation ω_o est rasante mais dans une direction où la surface anisotrope n'est pas rugueuse, il n'y a pas d'effets de visibilité. (d) La direction d'éclairage ω_i est rasante dans une direction où la surface anisotrope est rugueuse. Seule la couleur verte (aux sommets de la micro-surface) est éclairée et la couleur rouge est ombrée : la couleur moyenne visible et éclairée de la micro-surface est le vert foncé.

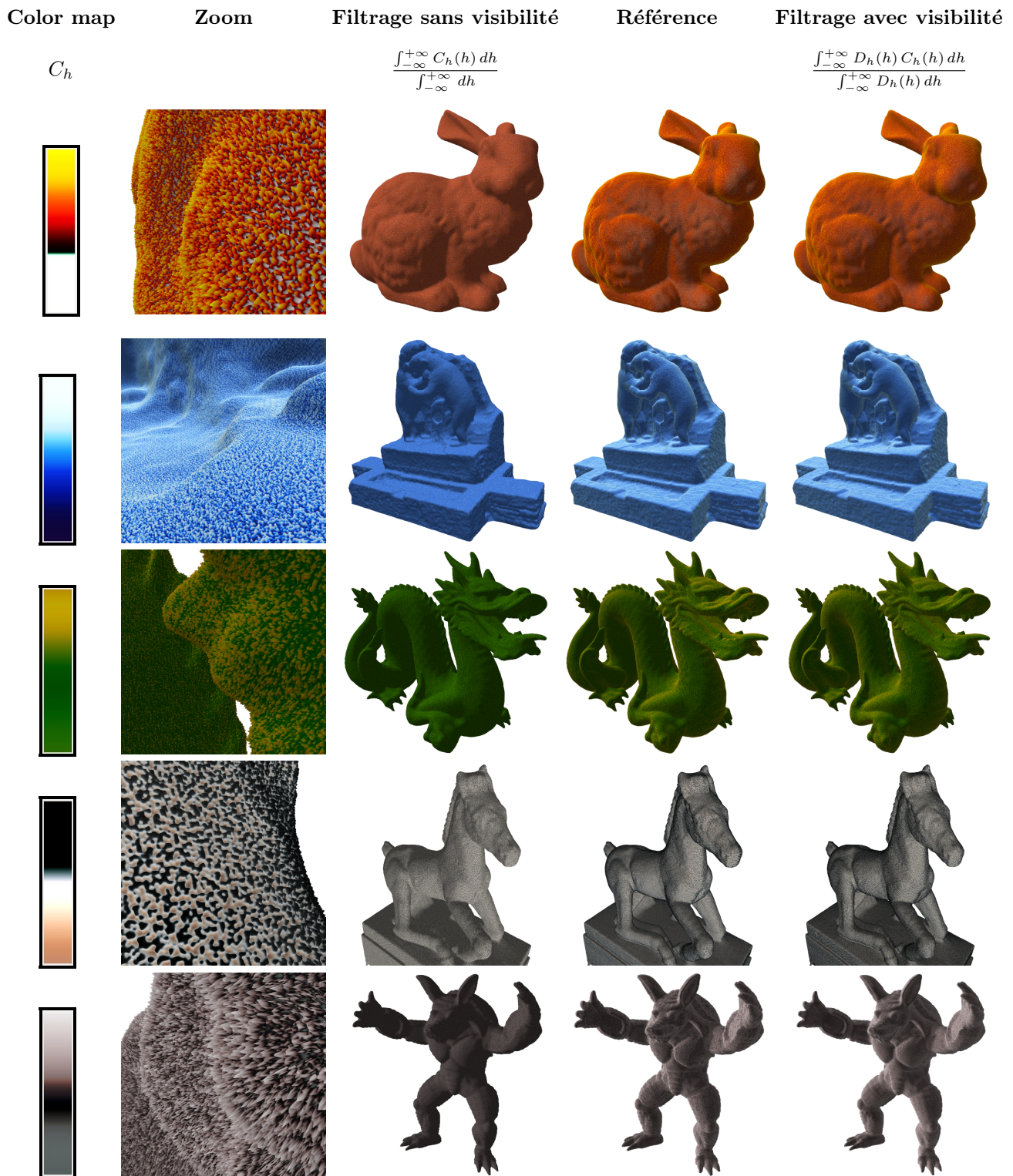


FIGURE 9.14 – Filtrage de color maps appliquées sur les hauteurs de la micro-surface. Ne pas tenir compte de la visibilité dans le filtrage produit une évaluation fautive de la couleur moyenne, en particulier dans les silhouettes où les effets dus à la visibilité sont importants. Les performances sont similaires pour les deux modèles de filtrage et dépendent essentiellement de la complexité du modèle géométrique (200Hz pour le bunny, 80Hz pour les ours). La référence est calculée par suréchantillonnage de la géométrie et du shading avec 256 échantillons par pixel.

9.4.3 Filtrage des color maps appliquées sur les normales

Matériau corrélé avec les normales. Dans cette section, nous cherchons à calculer la valeur \bar{C} du paramètre du matériau donné par l'équation (9.4) lorsque celui-ci est corrélé uniquement avec les **normales** de la micro-surface :

$$\begin{aligned}\bar{C} &= \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m} \\ &= \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle dp_m} \frac{\int_{\mathcal{M}} G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m}{\int_{\mathcal{M}} G_2^{\text{dist}}(\omega_o, \omega_i, p_m) dp_m} \\ &= \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle dp_m}.\end{aligned}\quad (9.9)$$

L'idée est de reformuler cette intégrale spatiale comme une intégrale statistique, cette fois-ci dans l'espace des normales de la surface.

Color map statistique des normales. Pour exprimer l'intégrale statistique en espace des hauteurs, nous devons exprimer la manière dont les paramètres de matériau C , se répartissent par hauteur et par normale. Cette répartition est donnée par la color map statistique des hauteurs et des normales, donnée par :

$$C_{\omega_m}(\omega_m') = \frac{\int_{\mathcal{M}} \delta_{\omega_m'}(\omega_m(p_m)) C(p_m) dp_m}{\int_{\mathcal{M}} \delta_{\omega_m'}(\omega_m(p_m)) dp_m}, \quad (9.10)$$

où la valeur $C_{\omega_m}(\omega_m)$ représente le paramètre de matériau moyen des micro-facettes de normale ω_m . Il est à noter que cette color map est *locale* : elle dépend du voisinage \mathcal{M} , et *statistique* : cela ne signifie pas que toutes les normales ω_m de la micro-surface aient nécessairement la même couleur $C_{\omega_m}(\omega_m)$, mais que $C_{\omega_m}(\omega_m)$ est la couleur moyenne sur les micro-facettes de normale ω_m .

Distribution des normales intrinsèques de la micro-surface. Les pentes de la micro-surface $(x_{\bar{m}}, y_{\bar{m}})$ sont à valeurs dans $]-\infty, +\infty[^2$ et leur distribution intrinsèque sur la surface est notée P^{22} . Nous rappelons que dans le chapitre 8 nous avons fait le choix de représenter des surfaces à statistiques gaussiennes. Cela détermine la distribution P^{22} : c'est la densité de probabilité d'une loi gaussienne $\mathcal{N}((\bar{x}_{\bar{m}}, \bar{y}_{\bar{m}}), \Sigma)$, dont les moments d'ordre 1 et 2 sont disponibles dans une texture MIPmappée ou calculables à la volée dans le cas d'une surface multi-échelles procédurale.

Intégration statistique de la color map appliquée sur les normales. En utilisant l'expression statistique dans l'espace des normales de la color map $C_{\omega_m}(\omega_m)$, l'intégrale spatiale de l'équation 9.9 se réécrit comme une intégrale statistique en espace des normales :

$$\bar{C} = \frac{\int_{\Omega} C_{\omega_m}(\omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m}{\int_{\Omega} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m}.$$

Cette intégrale peut être évaluée par intégration numérique, par exemple avec l'algorithme proposé dans le chapitre 10.

Intégration statistique par rapport à l'observateur uniquement. Il est possible de vouloir approximer cette intégrale uniquement en tenant compte de la corrélation avec le facteur de projection $\langle \omega_o, \omega_m \rangle$ pour la direction d'observation uniquement (par exemple lorsque l'éclairage est très basses-fréquences). Dans ce cas nous approximons :

$$\begin{aligned}\bar{C} &= \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle \langle \omega_i, \omega_m(p_m) \rangle dp_m} \\ &\approx \frac{\int_{\mathcal{M}} C(p_m) \langle \omega_o, \omega_m(p_m) \rangle dp_m}{\int_{\mathcal{M}} \langle \omega_o, \omega_m(p_m) \rangle dp_m}.\end{aligned}$$

et l'équation statistique devient :

$$\begin{aligned}\bar{C} &= \frac{\int_{\Omega} C_{\omega_m}(\omega_m) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m}{\int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m} \\ &= \int_{\Omega} C_{\omega_m}(\omega_m) D_{\omega_o}(\omega_m) d\omega_m,\end{aligned}\tag{9.11}$$

où :

$$D_{\omega_o}(\omega_m) = \frac{\langle \omega_o, \omega_m \rangle D(\omega_m)}{\int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m},$$

est la distribution de normales visibles introduite dans le chapitre 4.

Approximation gaussienne de la distribution des pentes visibles. Pour pouvoir calculer le produit scalaire $\int_{\Omega} C(\omega_m) D_{\omega_o}(\omega_m) d\omega_m$ présenté ci-dessus, nous formulons son intégrale dans l'espace des pentes qui offre une paramétrisation cartésienne commode pour le pré-filtrage :

$$\int_{\Omega} C(\omega_m) D_{\omega_o}(\omega_m) d\omega_m = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} C(x_{\tilde{m}}, y_{\tilde{m}}) P_{\omega_o}^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dx_{\tilde{m}} dy_{\tilde{m}},$$

où $P_{\omega_o}^{22}$ est la distribution des pentes visibles. Pour calculer ces produits scalaires, nous pré-filtrons C_{ω_m} avec un filtre gaussien et nous approximos $P_{\omega_o}^{22}$ avec une distribution gaussienne 2D paramétrée par ses moments, dans le même esprit que dans les sections précédentes. Nous utilisons les paramètres de la distribution gaussienne approximat $P_{\omega_o}^{22}$ pour paramétrer l'accès dans les tables pré-convoluées (nous utilisons un fetch hardware anisotrope lorsque la gaussienne 2D est anisotrope). La figure 9.15 illustre notre approximation de $P_{\omega_o}^{22}$ avec une distribution gaussienne 2D.

Les détails des dérivations pour le calcul de ces moments sont donnés dans l'annexe E.

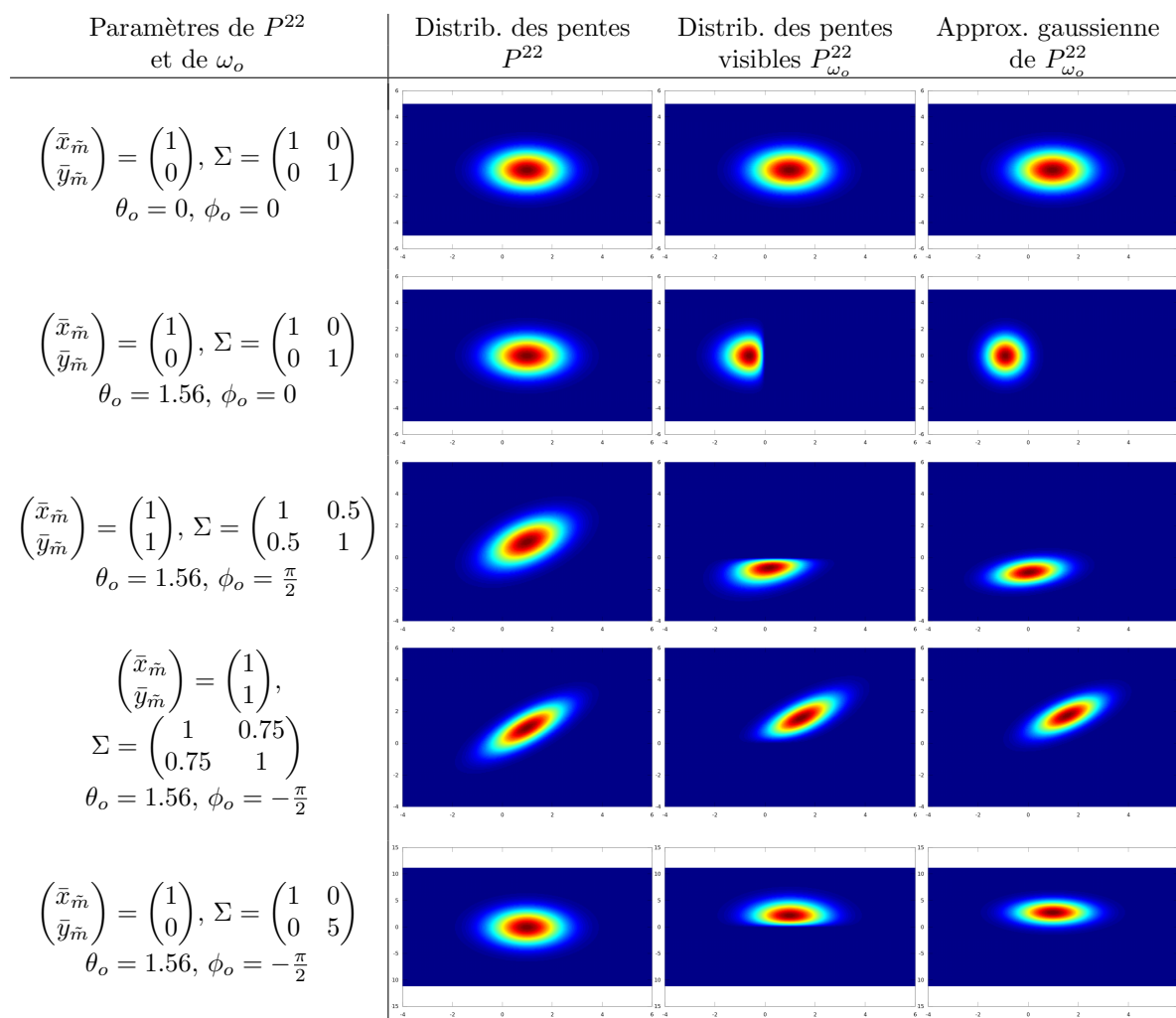


FIGURE 9.15 – Illustration de notre approximation gaussienne de la distribution des pentes visibles $P_{\omega_o}^{22}$ pour différentes configurations.

Résultats. La figure 9.16 illustre des résultats où nous filtrons des matériaux qui sont représentés comme une color map $C(\omega_m)$ appliquée sur les normales de la micro-surface et qui contient des valeurs RGB.

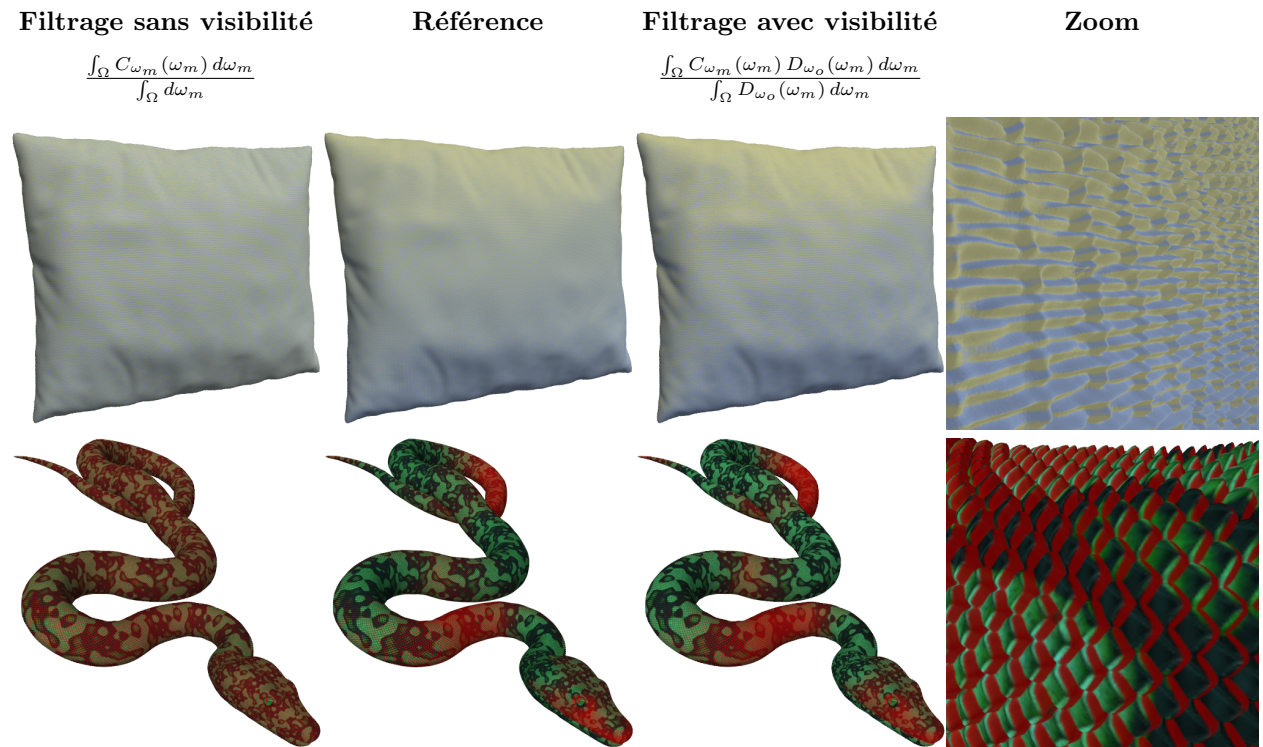


FIGURE 9.16 – Filtrage de color maps appliquées sur les normales de la micro-surface. Ne pas tenir compte de la visibilité dans le filtrage produit une évaluation fautive de la couleur moyenne, en particulier dans les silhouettes où les effets dus à la visibilité sont importants. Les performances sont similaires pour les deux modèles de filtrage et dépendent essentiellement de la complexité du modèle géométrique (130Hz pour le serpent). La référence est calculée par suréchantillonnage de la géométrie et du shading avec 256 échantillons par pixel.

9.4.4 Exprimer l'intégration de l'éclairage comme un problème de filtrage de color map

Le chapitre 10 est dédié au calcul de l'intégration de la BRDF avec l'éclairage dans le cas général. Dans cette section, nous allons voir un cas particulier où l'évaluation de la BRDF s'exprime comme un problème de color mapping sur les normales (ou les pentes) de la micro-surface.

Reformulation de l'équation d'illumination locale comme un problème de color mapping. La macro-BRDF est donnée par l'équation :

$$\rho(\omega_o, \omega_i) = \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}| |\omega_i \cdot \omega_{\bar{m}}|} \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m,$$

et la radiance sortante est donnée par :

$$\begin{aligned} L(\omega_o) &= \int_{\Omega_i} L(\omega_i) \rho(\omega_o, \omega_i) \langle \omega_i, \omega_{\bar{m}} \rangle d\omega_i \\ &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}|} \int_{\Omega_i} L(\omega_i) \int_{\Omega} \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) d\omega_m d\omega_i \\ &= \frac{|\omega_g \cdot \omega_{\bar{m}}|}{|\omega_o \cdot \omega_{\bar{m}}|} \int_{\Omega} G_1(\omega_o, \omega_m) \left(\int_{\Omega_i} L(\omega_i) \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_i, \omega_m \rangle \frac{G_2(\omega_o, \omega_i, \omega_m)}{G_1(\omega_o, \omega_m)} d\omega_i \right) \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m \\ &= \int_{\Omega} L(\omega_o, \omega_m) D_{\omega_o}(\omega_m) d\omega_m, \end{aligned} \quad (9.12)$$

où $D_{\omega_o}(\omega_m)$ est la distribution des normales visibles et

$$L(\omega_o, \omega_m) = \int_{\Omega_i} L(\omega_i) \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_i, \omega_m \rangle \frac{G_2(\omega_o, \omega_i, \omega_m)}{G_1(\omega_o, \omega_m)} d\omega_i,$$

est la radiance émise par les micro-facettes de normales ω_m . Cette dérivation montre que, d'une manière similaire à l'équation (9.11), l'évaluation de la radiance sortante $L(\omega_o)$ de la micro-surface est le produit scalaire entre la radiance par micro-facette $L(\omega_o, \omega_m)$ et la distribution de normales visibles $D_{\omega_o}(\omega_m)$.

Application au filtrage des irradiance maps. Nous pouvons considérer que la fonction $L(\omega_o, \omega_m)$ est, pour une direction d'observation ω_o fixée, une sorte de color map appliquée sur les normales :

$$L(\omega_o, \omega_m) = C(\omega_m).$$

Pour évaluer la radiance sortante $L(\omega_o)$, il faut donc savoir représenter cette color map et évaluer son produit scalaire avec la distribution des normales visibles D_{ω_o} . En supposant que la color map $C(\omega_m)$ est disponible et pré-intégrable, nous procédons de la même manière que dans la section précédente : nous approximons D_{ω_o} par une distribution gaussienne et nous l'utilisons pour accéder à une color map pré-convoluée, comme illustré dans l'algorithme de la figure 9.11.

Un exemple typique où la formulation basée sur une représentation de la radiance pré-intégrée peut s'appliquer en pratique est le rendu en temps réel des terrains dont le matériau de base $\rho_{\mathcal{M}} = \frac{a}{\pi}$ est une BRDF diffuse et où la fonction d'ombrage est négligée. En effet, dans ce cas nous avons :

$$L(\omega_o, \omega_m) = L(\omega_m) = \frac{a}{\pi} \int_{\Omega} L(\omega_i) \langle \omega_i, \omega_m \rangle d\omega_i,$$

qui est une fonction qui dépend uniquement des normales et non de la direction d'observation ω_o . Dans ce cas, la fonction $C(\omega_m)$ est une fonction 2D qui peut être facilement stockée telle quelle et pré-convoluée. Des résultats où cette représentation est utilisée sont donnés en figure 9.17.

Filtrage sans visibilité

$$\frac{\int_{\Omega} L(\omega_o, \omega_m) d\omega_m}{\int_{\Omega} d\omega_m}$$

Référence

Filtrage avec visibilité

$$\frac{\int_{\Omega} L(\omega_o, \omega_m) D_{\omega_o}(\omega_m) d\omega_m}{\int_{\Omega} D_{\omega_o}(\omega_m) d\omega_m}$$

Zoom

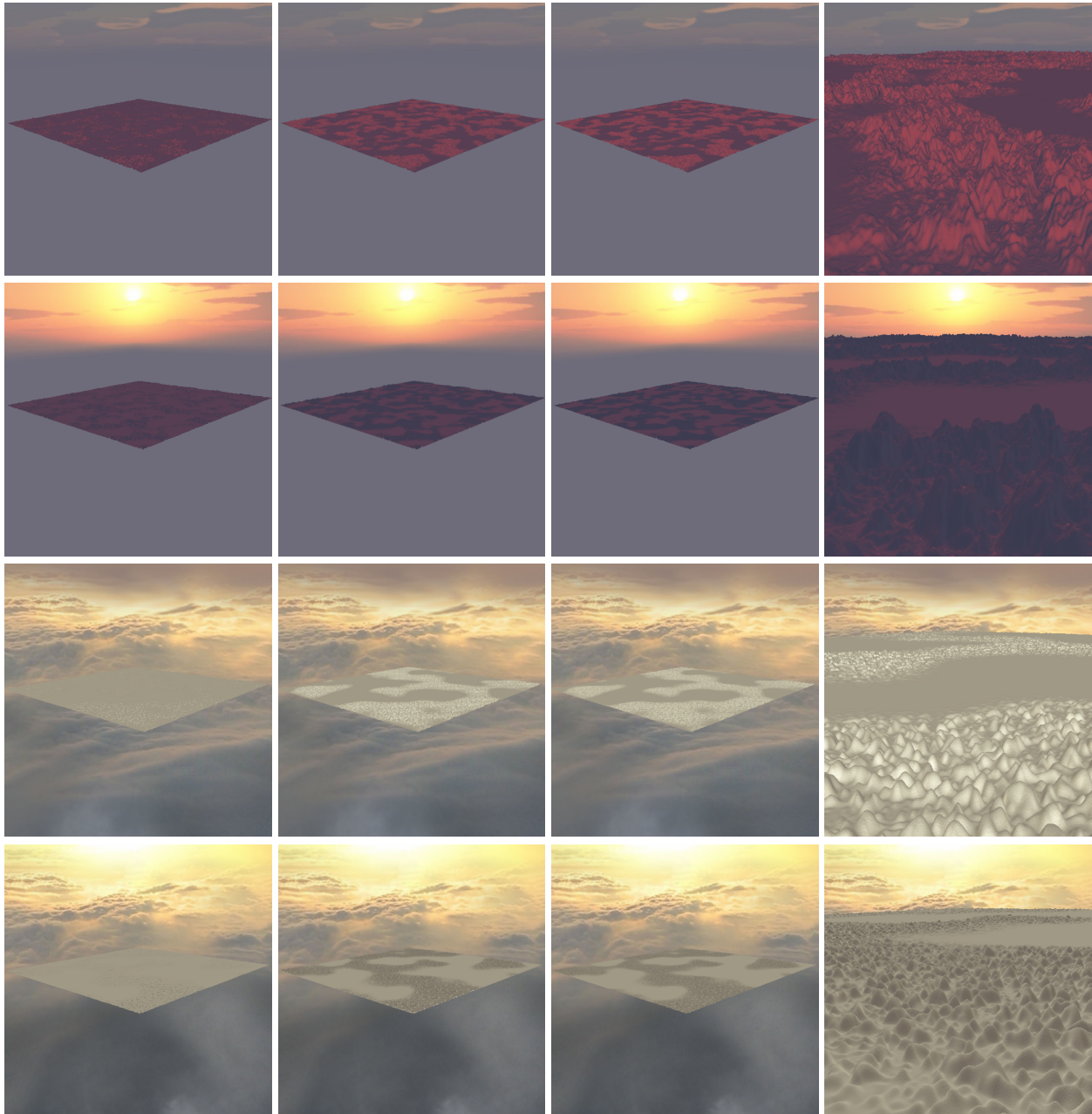


FIGURE 9.17 – Filtrage d'irradiance maps appliquées sur les normales de la micro-surface. La radiance sortante $L(\omega_o, \omega_m)$ est une fonction qui ne dépend que des normales ω_m . Lorsque la source de lumière la plus importante vient de la même direction que la direction d'observation ω_o (ligne 1 et 3), les normales orientées vers ω_o réfléchissent plus de lumière. Lorsque la source de lumière vient principalement de la direction opposée à la direction d'observation, les normales orientées vers ω_o réfléchissent peu de lumière. Notre shading filtré maintient ces effets dépendant du point de vue qui sont perdus avec un filtrage qui ne tient pas compte de la visibilité.

9.5 A propos des approximations gaussiennes

Les algorithmes de filtrage décrits dans ce chapitre se basent sur plusieurs approximations gaussiennes. Dans cette section, nous discutons leur validité et leur robustesse.

La figure 9.18 illustre différents types de surface, leurs distributions de pentes et l'approximation de leurs distributions de pentes par une gaussienne. Une implication du Théorème Central Limite est que plus la surface est déstructurée, stochastique ou faite de plusieurs couches additives, plus elle tend à avoir des statistiques gaussiennes. De fait, notre approximation est très précise dans de nombreux cas réels. En revanche, avec des surfaces répétitives, multi-modales ou structurées, l'approximation n'est pas très précise.

Analyse de l'erreur. La figure 9.18 illustre l'approximation intrinsèque à notre représentation gaussienne. Nous allons détailler la sensibilité du calcul à l'erreur introduite par cette approximation. Nous avons vu que le problème général du filtrage des color maps appliquées sur des textures ou des surfaces peut être résolu en calculant le produit scalaire entre la color map C et la distribution D de l'attribut générique a sur lequel la color map $C(a)$ est appliquée. Nous avons :

$$\bar{C} = \int C(a) D(a) da,$$

où a peut représenter différentes quantités, comme la valeur d'une fonction de bruit, une hauteur, une pente, etc. Il est important de noter que cette formulation, utilisée dans ce chapitre, n'est pas une approximation mais est exacte. Par contre, pour l'évaluer efficacement, en pratique, nous approximations D avec une gaussienne $G \approx D$ et nous pré-calculons et stockons les produits scalaires $\int C(a) G(a) da$ dans une color map pré-convoluée. Ce produit scalaire peut être exprimé dans le domaine fréquentiel, puisque la transformée de Fourier est un opérateur linéaire et préserve le produit scalaire :

$$\bar{C} = \int \mathcal{F}\{C\}(w) \mathcal{F}\{G\}(w) dw,$$

où \mathcal{F} est l'opérateur qui représente la transformée de Fourier. Remplacer D par G préserve la valeur $\mathcal{F}\{G\}(0)$ du spectre à la DC (en 0). En effet, la fréquence nulle est la valeur moyenne de G , et donc de D , puisque G est paramétrée par les moments de D dans notre algorithme. En général, les basses fréquences avec des périodes bien plus larges que la déviation standard de la distribution sont préservées en remplaçant D par G . Une propriété spectrale importante de l'approximation gaussienne est donc que l'erreur se trouve d'abord dans les plus hautes fréquences. Cela implique que, étant donné le produit scalaire entre C et G , plus le contenu fréquentiel de C est élevé, plus la sensibilité à l'erreur introduite par l'approximation gaussienne est grande. La figure 9.19 illustre ce comportement en appliquant des color maps sur les pentes de surfaces avec des statistiques parfaitement gaussiennes ou non : le résultat est précis même avec des surfaces non gaussiennes lorsque la color map est très basses-fréquences. Notre technique est donc particulièrement adaptée à l'usage des color maps représentées par des fonctions linéaires puisqu'elles ont un spectre nul en dehors de la DC. Si la color map est linéaire, ou affine, c'est-à-dire si $C(a) = c_1 a + c_2$, alors :

$$\bar{C} = \int C(a) D(a) da = c_1 \int a D(a) da + c_2,$$

où $\int a D(a) da$ est le moment d'ordre 1 de la distribution (la position moyenne) et il est préservé par notre approximation gaussienne. A noter que même si la color map est linéaire, le problème de filtrage reste non linéaire car la distribution D est dépendante du point de vue. En effet, même si la color map est linéaire, la fonction à filtrer est le produit non linéaire entre la color map et la visibilité. C'est la raison pour laquelle un filtrage en espace texture qui ne tient pas compte de la visibilité produit un résultat incorrect lorsque la color map est appliquée sur un attribut corrélé à la visibilité, même lorsque celle-ci est linéaire.

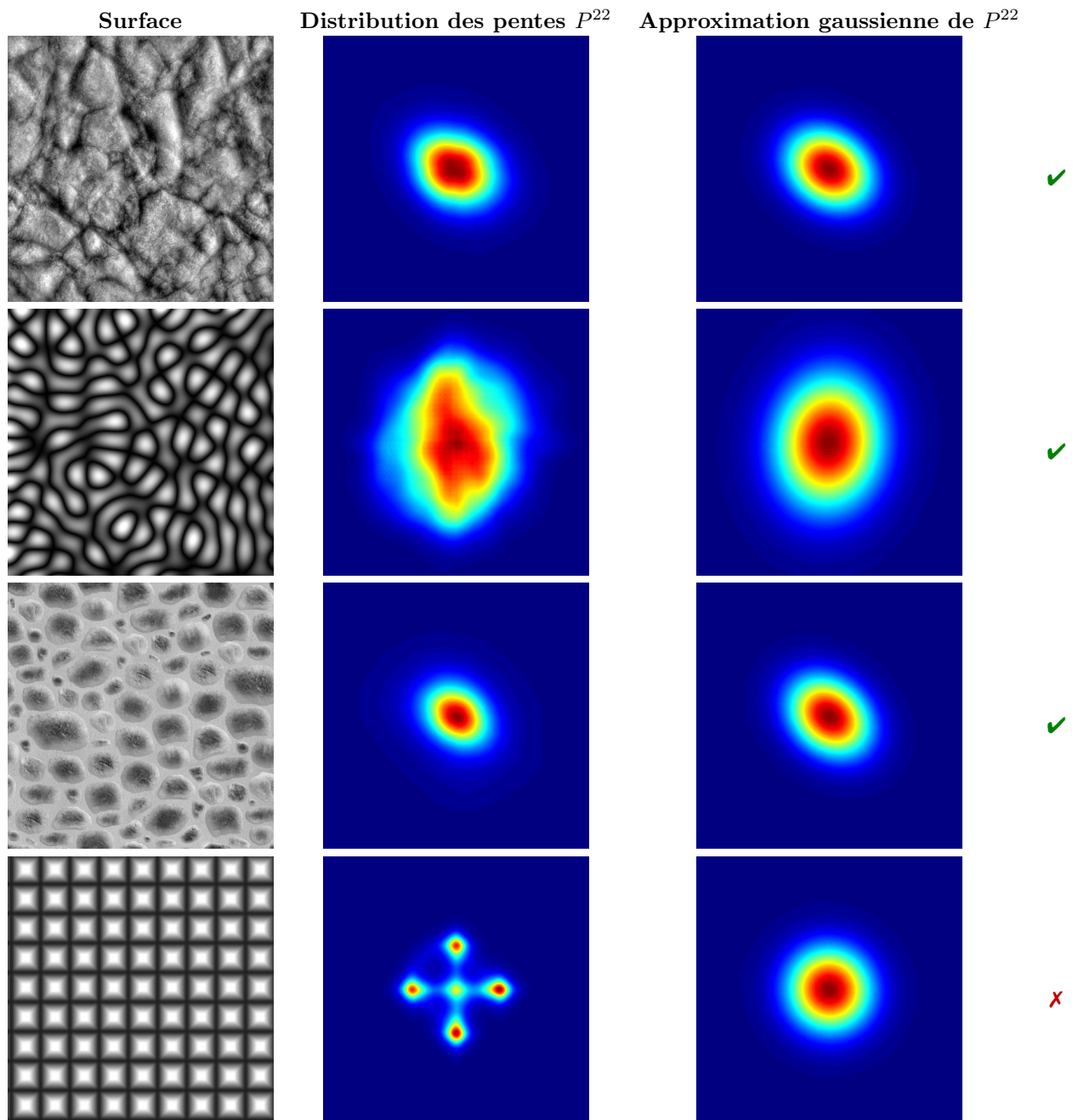


FIGURE 9.18 – Des micro-surfaces, leurs distributions de pentes et nos approximations gaussiennes associées. Les surfaces naturelles sont souvent bien approximées par des distributions gaussiennes tandis que les surfaces artificielles présentent des structures plus complexes.

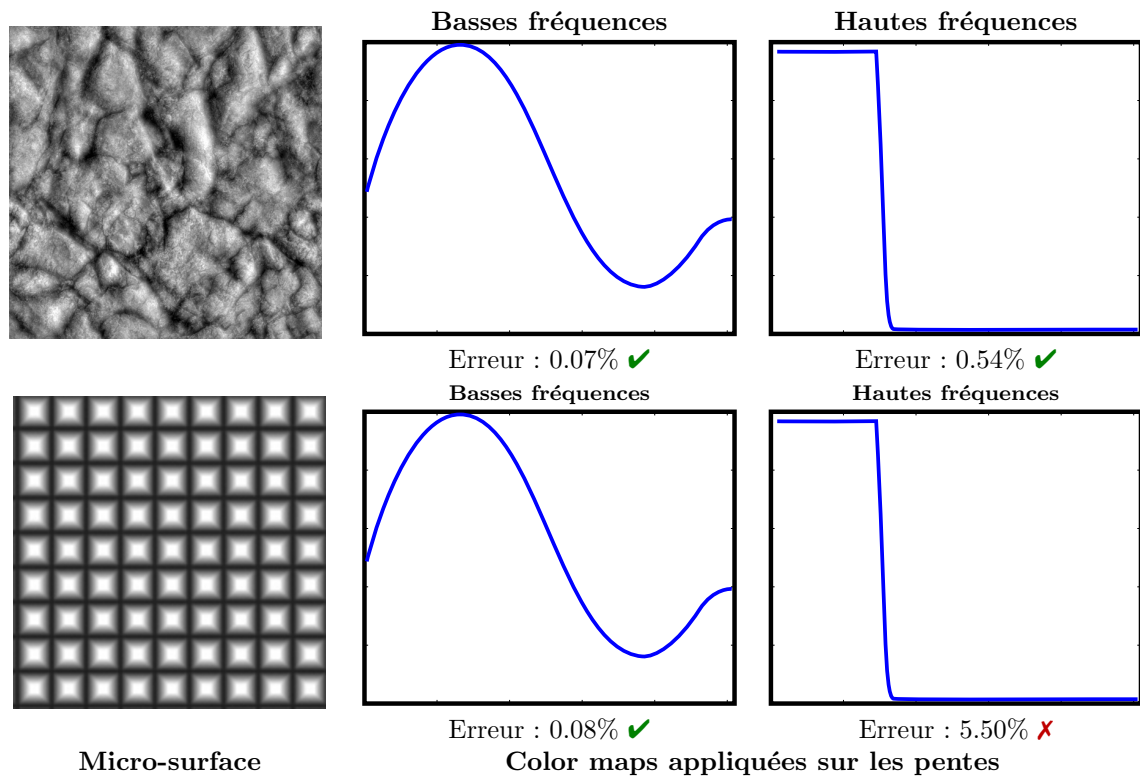


FIGURE 9.19 – Erreurs introduites par l'approximation gaussienne dans le filtrage des color maps appliquées sur les pentes. La color map 1D est appliquée sur la composante $x_{\bar{m}}$ des pentes. (haut) La distribution des pentes de la surface est bien approximée par une gaussienne et l'erreur est faible ($< 1\%$) pour les hautes fréquences comme pour les basses fréquences. (bas) La distribution des pentes s'approxime mal par une gaussienne. L'erreur est toujours négligeable avec une color map basses-fréquences mais devient importante ($> 5\%$) avec une color map hautes-fréquences.

9.6 Résumé du chapitre

Points clés du chapitre

- Dans une application de rendu de surface multi-échelles, le matériau de la surface est potentiellement variable contrairement à ce qui est supposé dans le modèle à micro-facettes classiques.
- Pour représenter correctement l'apparence multi-échelles de la surface, il faut savoir filtrer ce matériau en tenant compte de ses corrélations avec la visibilité depuis les directions ω_o et ω_i .
- Pour être filtrable, les paramètres du matériau doivent être représentés sous une forme linéairement combinable et séparable. Par commodité, nous appelons “couleur” ou “color map” un tel paramètre générique C .
- La représentation des corrélations entre les paramètres C du matériau et la visibilité peut se ramener à la définition de color maps $C(\omega_m, h)$ appliquées sur les hauteurs et les normales statistiques de la micro-surface.
- Pour filtrer ces color maps, il faut calculer le produit scalaire entre la color map et la distribution statistique des valeurs sur lesquelles elles sont appliquées.
- Lorsque les valeurs de ces fonctions sont corrélées à la visibilité, ce qui est le cas des hauteurs et des normales, il faut calculer le produit scalaire entre la color map et la distribution statistique des valeurs *visibles*.
- En pratique, nous réalisons ce calcul en approximant les distributions visibles des pentes et des hauteurs de la micro-surface avec des distributions gaussiennes, dont nous calculons les moments de manière analytique.

A voir

- L'annexe E propose la dérivation complète pour le calcul des moments de la distribution de pentes visibles ainsi que le code C++ associé.
- Les articles [[HNPN13a](#), [HNPN13b](#)], leurs supplemental materials, leurs vidéos et les slides de la présentation I3D.

Chapitre 10

Evaluation de la BRDF et de l'éclairage en temps-réel

Dans les chapitres précédents, nous avons défini les représentations multi-échelles pour la surface et pour ses matériaux. Nous avons vu comment ces représentations se combinent en une représentation unifiée : la macro-BRDF. Mais pour obtenir une image, c'est-à-dire calculer la radiance mesurée par le pixel, il ne suffit pas de connaître le matériau, il faut aussi calculer comment il interagit avec l'éclairage de la scène. L'objet de ce chapitre est de proposer une manière de calculer l'intégration de la BRDF du matériau avec différents types d'éclairage tout en respectant un cahier des charges temps-réel. Dans la section 10.1 nous développons les équations à calculer et dans la section 10.2 nous proposons des techniques d'intégration numérique pour les évaluer efficacement. Dans la section 10.3, nous commenterons les résultats obtenus avec ces techniques et dans la section 10.4 nous discuterons certains problèmes liés aux hypothèses du modèle.

10.1 Les équations d'illumination locale à évaluer

Dans cette section, nous rappelons les différents éléments qui interviennent dans les équations que nous cherchons à calculer. Le calcul de la radiance arrivant dans un pixel et qui décrit la manière dont le matériau interagit avec l'éclairage est donné par l'équation d'illumination locale :

$$L(\omega_o) = \int_{\Omega} L(\omega_i) \rho(\omega_o, \omega_i) \langle \omega_g, \omega_i \rangle d\omega_i,$$

où ρ est la BRDF (le matériau) et L la radiance incidente (l'éclairage). Pour instancier cette équation, nous définissons ces deux fonctions en pratique : nous choisissons des modèles de BRDFs et des types d'éclairage.

10.1.1 Modèles de BRDFs

Dans les chapitres 8 et 9, nous avons choisi de représenter la BRDF dans chaque canal (pour chaque fréquence) par une distribution décentrée de micro-facettes spéculaires et/ou diffuses P^{22} avec un profil de micro-surface de Smith. Nous donnons les équations pour ces deux modèles de BRDFs.

BRDF à micro-facettes spéculaires. Lorsque les micro-facettes sont spéculaires, l'équation d'illumination locale est donnée par :

$$L(\omega_o) = \frac{\omega_{\bar{m}} \cdot \omega_g}{\omega_{\bar{m}} \cdot \omega_o} \int_{\Omega} \frac{L(\omega_i) F(\omega_h, \omega_i) \langle \omega_m, \omega_o \rangle}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} D(\omega_m) d\omega_m.$$

En reformulant cette intégrale dans l'espace des pentes avec le changement de variable

$D(\omega_m) d\omega_m = \frac{P^{22}(x_{\bar{m}}, y_{\bar{m}})}{\omega_m \cdot \omega_g} dx_{\bar{m}} dy_{\bar{m}}$, nous obtenons :

$$L(\omega_o) = \frac{\omega_{\bar{m}} \cdot \omega_g}{\omega_{\bar{m}} \cdot \omega_o} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{L(\omega_i) F(\omega_h, \omega_i) \frac{\langle \omega_m, \omega_o \rangle}{\omega_m \cdot \omega_g}}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} P^{22}(x_{\bar{m}}, y_{\bar{m}}) dx_{\bar{m}} dy_{\bar{m}}. \quad (10.1)$$

BRDF à micro-facettes diffuses. Lorsque les micro-facettes sont diffuses, l'équation d'illumination locale est donnée par :

$$L(\omega_o) = \frac{\omega_{\bar{m}} \cdot \omega_g}{\omega_{\bar{m}} \cdot \omega_o} \int_{\Omega} L(\omega_i) \frac{\frac{1}{\pi} \int_{\Omega} \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} d\omega_i.$$

En reformulant cette intégrale dans l'espace des pentes avec le changement de variable

$D(\omega_m) d\omega_m = \frac{P^{22}(x_{\bar{m}}, y_{\bar{m}})}{\omega_m \cdot \omega_g} dx_{\bar{m}} dy_{\bar{m}}$, nous obtenons :

$$L(\omega_o) = \frac{\omega_{\bar{m}} \cdot \omega_g}{\omega_{\bar{m}} \cdot \omega_o} \int_{\Omega} L(\omega_i) \frac{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle}{\langle \omega_m, \omega_g \rangle} P^{22}(x_{\bar{m}}, y_{\bar{m}}) dx_{\bar{m}} dy_{\bar{m}}}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} d\omega_i. \quad (10.2)$$

10.1.2 Modèles d'éclairages

Dans les applications temps-réel, typiquement deux types d'éclairage sont utilisés : les sources ponctuelles/directionnelles et l'éclairage environnemental.

Eclairage ponctuel/directionnel. Lorsque la source de lumière est ponctuelle ou directionnelle, elle se note $L(\omega'_i) = L_i \delta_{\omega_i}(\omega'_i)$, où ω_i est sa direction depuis la surface (considérée constante dans le voisinage \mathcal{M}) et L_i sa puissance. Dans ce cas, l'évaluation de la BRDF à micro-facettes spéculaires a une forme close analytique que nous évaluons directement :

$$L(\omega_o) = \frac{\omega_{\bar{m}} \cdot \omega_g}{\omega_{\bar{m}} \cdot \omega_o} \frac{L_i F(\omega_h, \omega_i)}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} D(\omega_h), \quad (10.3)$$

et l'évaluation de la BRDF à micro-facettes diffuses s'exprime comme une intégrale dans l'espace des pentes :

$$L(\omega_o) = L_i \frac{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle}{\langle \omega_m, \omega_g \rangle} P^{22}(x_{\bar{m}}, y_{\bar{m}}) dx_{\bar{m}} dy_{\bar{m}}}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)}. \quad (10.4)$$

Eclairage environnemental. Pour représenter de l'éclairage environnemental, nous utilisons des *cube maps* qui représentent la radiance incidente $L(\omega_i)$.

Irradiance mapping. Dans de nombreuses applications de rendu en temps-réel, la partie diffuse de la BRDF est rendue en utilisant une *irradiance map* $E(\omega_m)$, qui représente l'irradiance totale arrivant depuis l'environnement sur une surface de normale ω_m . Formellement, l'équation 10.2 de la BRDF à micro-facettes diffuses peut s'écrire :

$$L(\omega_o) = \frac{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} E(\omega_m) \frac{\langle \omega_o, \omega_m \rangle}{\langle \omega_m, \omega_g \rangle} P^{22}(x_{\bar{m}}, y_{\bar{m}}) dx_{\bar{m}} dy_{\bar{m}}}{1 + \Lambda(\omega_o)}, \quad (10.5)$$

où l'irradiance map est définie dans le voisinage \mathcal{M} par :

$$E(\omega_m) = \int_{\Omega} L(\omega_i) \langle \omega_i, \omega_m \rangle \frac{1 + \Lambda(\omega_o)}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} d\omega_i. \quad (10.6)$$

Cette reformulation est exacte mais elle présente l'inconvénient que la définition de l'irradiance map E dépend du voisinage \mathcal{M} , qui intervient à travers la fonction $\frac{1+\Lambda(\omega_o)}{1+\Lambda(\omega_o)+\Lambda(\omega_i)}$ qui représente l'ombrage moyen visible sur la micro-surface du voisinage \mathcal{M} . En pratique, il faudrait calculer une fonction $E(\omega_m)$ par voisinage \mathcal{M} , ce qui n'est pas toujours raisonnable en terme de budget mémoire et de temps de calcul. Pour avoir une représentation générale et utilisable indépendamment du voisinage, le facteur d'ombrage est généralement négligé et l'irradiance map est approximée par :

$$E(\omega_m) = \int_{\Omega} L(\omega_i) \langle \omega_i, \omega_m \rangle d\omega_i. \quad (10.7)$$

Une telle irradiance map représente la radiance incidente pré-convoluée avec le produit scalaire $\langle \omega_i, \omega_m \rangle$. Cette fonction se représente efficacement en pratique avec les 9 premières harmoniques sphériques car elle est très basses-fréquences [RH01].

10.2 Evaluer les équations par intégration numérique

Dans la section précédente, nous avons reformulé le calcul de l'équation d'illumination locale comme des intégrales dans l'espace des pentes. En particulier, les équations (10.1), (10.4) et (10.5) ont la particularité de toutes s'exprimer sous la forme :

$$L(\omega_o) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x_{\tilde{m}}, y_{\tilde{m}}) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dx_{\tilde{m}} dy_{\tilde{m}}, \quad (10.8)$$

où f est une fonction à intégrer qui dépend de la pente $(x_{\tilde{m}}, y_{\tilde{m}})$ (ou de la normale ω_m). Nous proposons une méthodologie générale pour calculer numériquement des intégrales exprimées sous cette forme avec un nombre constant d'échantillons, c'est-à-dire avec un temps de calcul constant, critère indispensable pour garantir des performances temps-réel.

10.2.1 Discrétisation des intégrales

Discrétisation de l'intégrale. Pour évaluer numériquement les intégrales sous la forme de celle de l'équation (10.8), nous les approximons par une somme discrète :

$$L(\omega_o) \approx \sum_{j=1}^N \sum_{k=1}^N f(x_{\tilde{m}}^{j,k}, y_{\tilde{m}}^{j,k}) W_{\tilde{m}}^{j,k},$$

où les positions $((x_{\tilde{m}}^{j,k}, y_{\tilde{m}}^{j,k}))$ et les poids $W_{\tilde{m}}^{j,k}$ des échantillons sont calculés en fonction de la distribution P^{22} .

Calcul de la grille d'échantillonnage. Dans notre méthode, les échantillons sont dans un premier temps distribués uniformément dans le plan \mathbb{R}^2 de l'espace des pentes. Pour un ensemble 2D de $N \times N$ échantillons, nous n'avons besoin de ne stocker que $\lceil \frac{N}{2} \rceil$ positions et poids. Nous initialisons chaque échantillon en échantillonnant uniformément une densité de loi normale 1D $\mathcal{N}(0, 1)$ de manière à ce que l'ensemble des échantillons recouvre un intervalle arbitraire $[-\lambda, \lambda] \times [-\lambda, \lambda]$, choisi pour maximiser la précision du calcul (Figure 10.1(a)). Par exemple, nous choisissons $\lambda = 1.8$ pour $N = 5$. Les poids normalisés W_j associés aux points p_j sont donnés par :

$$W_j = \frac{\exp(-\frac{1}{2} p_j^2)}{\sum_{k=1}^N \exp(-\frac{1}{2} p_k^2)}.$$

Au moment du rendu, nous remplaçons les échantillons en fonction des paramètres de la distribution P^{22} à échantillonner. Cela nous donne de nouvelles positions d'échantillonnage $\tilde{n} = (x_{\tilde{m}}, y_{\tilde{m}})$ et des poids $W_{\tilde{m}}$.

Pour réaliser cela en pratique, nous multiplions les échantillons 2D (p_j, p_k) par la matrice obtenue en faisant une décomposition de Cholesky sur la matrice de covariance σ de P^{22} [Jäc05] :

$$\begin{aligned}x_{\tilde{m}}^{j,k} &= p_j \sigma_x + \mathbb{E}[x_{\tilde{m}}], \\y_{\tilde{m}}^{j,k} &= \left(r_{xy} p_j + \sqrt{1 - r_{xy}^2} p_k \right) \sigma_y + \mathbb{E}[y_{\tilde{m}}], \\W_{\tilde{m}}^{j,k} &= W_j W_k,\end{aligned}$$

où $r_{xy} = c_{xy}/(\sigma_x \sigma_y)$ est le coefficient de corrélation de Σ . Cet algorithme est illustré dans la figure 10.1, et produit une collection d'échantillons faciles à calculer. Cette méthode d'intégration converge vers la valeur exacte lorsque la taille N de la grille d'échantillonnage augmente (preuve : c'est la définition même de l'intégrale au sens de Riemann).

Scalabilité de la technique. Le problème d'intégration de la radiance sur la micro-surface est un problème d'intégration spatiale non scalable : l'intégrale à calculer dépend de la taille du domaine d'intégration (la taille de la micro-surface). En réalisant une intégration dans l'espace des pentes, nous nous ramenons à un problème d'intégration statistique scalable : le nombre d'échantillons utilisés pour calculer l'intégrale ne dépend ni de la taille ni des statistiques de la micro-surface, la complexité de l'algorithme d'intégration numérique est donc constante.

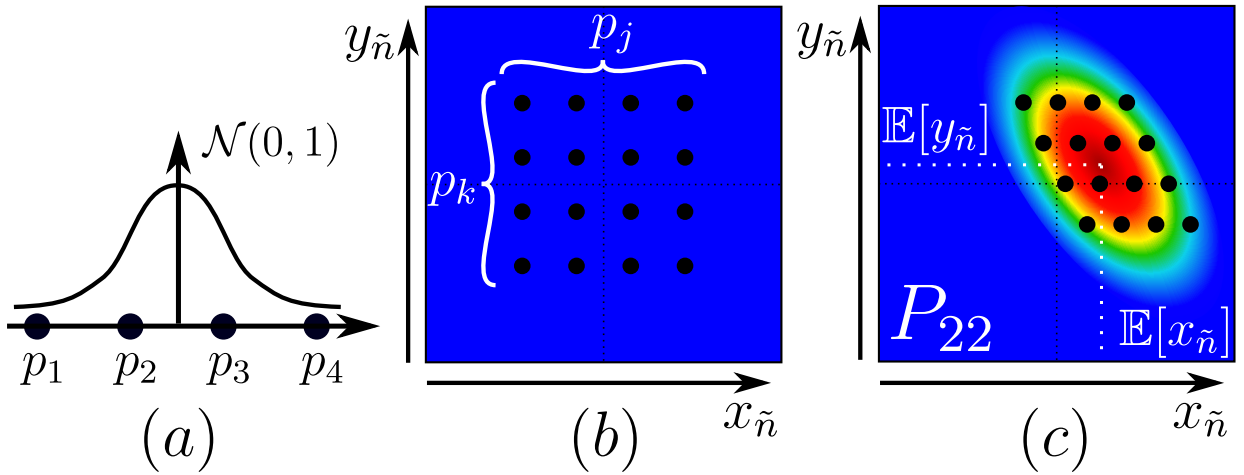


FIGURE 10.1 – Construction des échantillons en fonction de P^{22} . (a) Nous pré-calculons des échantillons 1D p_j . (b) Dans le fragment shader, nous calculons le produit cartésien de l'ensemble des échantillons par lui-même et obtenons un ensemble d'échantillons 2D (p_j, p_k) . (c) Nous transformons les positions des échantillons en fonction de la moyenne et de la matrice de covariance de P^{22} .

10.2.2 Intégration de la BRDF à micro-facettes diffuses

Problème. Contrairement à la BRDF à micro-facettes spéculaires, l'évaluation de la BRDF à micro-facettes diffuses n'a pas de solution analytique et nécessite une intégration numérique, comme le montre l'équation (10.4). Cela pose un problème de performances lorsque le nombre de sources ponctuelles à évaluer est important, ce qui est le cas par exemple dans certains jeux vidéos où l'illumination globale est calculée en utilisant un grand nombre de *Virtual Point Lights* [Kel97]. En effet, si nous utilisons un schéma d'intégration numérique à N échantillons, alors évaluer un nombre L de sources ponctuelles a une complexité algorithmique en $O(LN)$. Pour remédier à ce problème, nous proposons un algorithme en deux passes : nous commençons par calculer une irradiance map avec l'ensemble des sources de lumière, puis nous intégrons la BRDF avec l'irradiance map en une seule fois. La construction de l'irradiance map est en $O(L)$ et l'intégration numérique est en $O(N)$. En utilisant cette technique nous remplaçons l'algorithme avec une complexité en produit $O(LN)$ par un algorithme avec une complexité en somme $O(L) + O(N)$.

Construction de l'irradiance map. La première étape consiste à construire une irradiance map à partir de l'éclairage de la scène, c'est-à-dire les sources ponctuelles et/ou directionnelles et l'éclairage environnemental. Puisque l'irradiance map $E(\omega_m)$ définie par l'équation (10.6) dépend de la fonction de masquage et d'ombrage, nous la calculons à la volée pour le voisinage \mathcal{M} . Nous utilisons une représentation basée sur des coefficients d'harmoniques sphériques, particulièrement bien adaptée aux irradiance maps [RH01]. Dans l'algorithme 1, nous calculons les neuf premiers coefficients d'harmoniques sphériques de l'irradiance map E donnée par les sources ponctuelles et/ou directionnelles pondérées par la fonction de masquage et d'ombrage $\frac{1}{1+\Lambda(\omega_o)+\Lambda(\omega_i)}$ associée à leurs directions ω_i respectives. En revanche, calculer de manière interactive ces coefficients pour une cube map complète n'est pas possible car ce serait trop coûteux. Pour l'éclairage environnemental représenté par une cube map, nous les calculons une seule fois sur le CPU en utilisant la formulation simplifiée de l'équation 10.7 qui néglige le facteur d'ombrage. Avec cette approche, nous pondérons correctement l'éclairage pour des sources ponctuelles et/ou directionnelles, mais nous négligeons l'ombrage dans l'éclairage environnemental. L'algorithme 1 donne notre procédure de calcul pour la construction de l'irradiance map.

Algorithm 1 Construction de l'irradiance map

```

function INITIRRADIANCEMAPCOEFS
  for  $0 \leq l \leq 2$  and  $-l \leq m \leq l$  do
     $L_{lm} = 0$ 
  end for
  for each directional light of radiance  $L$  and direction  $\delta_{\omega_i}$  do
     $P = L / (1 + \Lambda(\omega_o) + \Lambda(\omega_i))$ 
     $L_{00} += 0.282095 \times P$ 
     $(L_{11}, L_{10}, L_{1-1}) += 0.488603 \times (x_i, z_i, y_i) \times P$ 
     $(L_{21}, L_{2-1}, L_{2-2}) += 1.092548 \times (x_i z_i, y_i z_i, x_i y_i) \times P$ 
     $L_{20} += 0.315392 \times (3z_i^2 - 1) \times P$ 
     $L_{22} += 0.546274 \times (x_i^2 - y_i^2) \times P$ 
  end for
   $L_{lm} += L_{lm}(\text{environment}) / (1 + \Lambda(\omega_o))$ 
end function

```

Intégration de l'irradiance map. Une fois que nous avons calculé les coefficients de l'irradiance map, nous discrétisons l'équation 10.6 avec la méthode donnée dans la section 10.2 :

$$I = \frac{\omega_{\tilde{m}} \cdot \omega_g}{\omega_{\tilde{m}} \cdot \omega_o} \frac{1}{\pi} \sum_{N^2} E(\omega_m) \frac{\langle \omega_m, \omega_o \rangle}{\omega_m \cdot \omega_g} W_{\tilde{n}}. \quad (10.9)$$

Nous rappelons que l'évaluation de l'irradiance map se fait avec la formule suivante [RH01] :

$$\begin{aligned} E(\omega_m) = & 0.429043L_{22}(x_n^2 - y_n^2) + 0.743125L_{20}z_n^2 \\ & + 0.886227L_{00} - 0.247708L_{20} \\ & + 0.858086(L_{2-2}x_ny_n + L_{21}x_nz_n + L_{2-1}y_nz_n) \\ & + 1.023328(L_{11}x_n + L_{1-1}y_n + L_{10}z_n). \end{aligned}$$

L'algorithme 2 illustre l'évaluation de l'équation (10.9) en pratique, avec le placement des échantillons dans l'espace des pentes, la construction des normales et l'évaluation de l'irradiance map.

Algorithm 2 Echantillonnage de l'irradiance map

```

function IRRADIANCEMAPSAMPLING( $\omega_o, \mathbb{E}[\tilde{m}], \sigma_x, \sigma_y, c_{xy}$ )
   $I = 0$ 
  for  $j, k = 1..N$  do
     $x_{\tilde{m}} = p_j \sigma_x + \mathbb{E}[x_{\tilde{m}}]$ 
     $y_{\tilde{m}} = (r_{xy} p_j + \sqrt{1 - r_{xy}^2} p_k) \sigma_y + \mathbb{E}[y_{\tilde{m}}]$ 
     $W_{\tilde{m}} = W_j W_k$ 
     $\omega_m = (-x_{\tilde{m}}, -y_{\tilde{m}}, 1) / \sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}$ 
     $I += W_{\tilde{m}} \times \frac{\langle \omega_m, \omega_o \rangle}{\omega_m \cdot \omega_g} \times E(\omega_m)$ 
  end for
return  $I \times \frac{\omega_{\tilde{m}} \cdot \omega_g}{\omega_{\tilde{m}} \cdot \omega_o} \times \frac{1}{\pi}$ 
end function

```

10.2.3 Intégration de la BRDF à micro-facettes spéculaires

Problème. Nous avons vu que l'évaluation de la BRDF à micro-facettes spéculaires avec une seule direction d'éclairage incident est donnée de manière analytique par l'équation (10.3) et elle est donc facile à évaluer. En revanche, l'équation 10.1 qui donne l'évaluation de la BRDF avec un éclairage environnemental n'a pas de solution analytique et doit être intégrée numériquement. Avec la méthode donnée dans la section 10.2, nous discrétisons l'équation 10.1 et nous obtenons :

$$I = \frac{\omega_{\tilde{m}} \cdot \omega_g}{\omega_{\tilde{m}} \cdot \omega_o} \sum_{N^2} \frac{L(\omega_i) F(\omega_h, \omega_i) \frac{\langle \omega_m, \omega_o \rangle}{\omega_m \cdot \omega_g}}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} W_{\tilde{n}}. \quad (10.10)$$

Le problème principal de ce mode d'évaluation est que, contrairement à l'irradiance map $E(\omega_m)$ dans l'équation (10.9), l'éclairage incident $L(\omega_i)$ que nous cherchons à intégrer ici peut potentiellement contenir de très hautes fréquences. Un schéma d'intégration numérique doit donc utiliser une fréquence d'échantillonnage très élevée (donc une taille de grille N très grande) pour évaluer cette intégrale sans introduire d'aliasing, ce qui invalide la condition de scalabilité.

Echantillonnage pré-filtré. Pour ne pas introduire d'aliasing tout en restant scalable, nous utilisons un schéma d'intégration pré-filtrée inspiré de *filtered importance sampling* [CK07, KC08]. Cette technique est dans le même esprit que le rendu multi-échelles pré-filtré dont nous discutons dans cette thèse, sauf qu'ici ce ne sont pas les données de la surface qui sont pré-filtrées, mais les données de l'éclairage incident $L(\omega_i)$. Nous évaluons la somme :

$$I = \frac{\omega_{\tilde{m}} \cdot \omega_g}{\omega_{\tilde{m}} \cdot \omega_o} \sum_{N^2} \frac{[L * f](\omega_i) F(\omega_h, \omega_i) \frac{\langle \omega_m, \omega_o \rangle}{\omega_m \cdot \omega_g}}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} W_{\tilde{n}}, \quad (10.11)$$

où dans l'équation (10.10) nous avons remplacé la fonction d'éclairage incident $L(\omega_i)$ par une fonction d'éclairage incident pré-filtrée $[L * f](\omega_i)$ où f est une fonction filtre passe-bas qui supprime les hautes-fréquences. Comme nous l'avons expliqué dans le chapitre 1, la fonction filtre doit être choisie en fonction de la fréquence d'échantillonnage, c'est-à-dire en fonction de la période λ entre les échantillons. En pratique, nous utilisons une cube map pré-filtrée avec des niveaux de MIPmap qui correspondent à différentes tailles de filtre f , et nous choisissons, pour chaque échantillon \tilde{m} , la taille de filtre α en angle solide :

$$\alpha \approx \left\| \frac{\partial \omega_i}{\partial \tilde{m}} \right\| A,$$

où $A = (2\lambda \max(\sigma_x, \sigma_y)/N)^2$ est la surface recouverte par un échantillon dans l'espace des pentes. Multiplier A par le jacobien $\left\| \frac{\partial \omega_i}{\partial \tilde{m}} \right\|$ de la transformation des pentes \tilde{m} en directions réfléchies ω_i permet de convertir la surface recouverte par un échantillon dans l'espace des pentes en la surface recouverte par ce même échantillon en angle solide, dans l'espace des directions réfléchies. Comme nous utilisons des échantillons isotropes, nous surestimons cette surface en la calculant à partir de la plus grande valeur propre de P^{22} , d'où le terme $\max(\sigma_x, \sigma_y)$. Cela garantit que, pour n'importe quelle valeur de N , les filtres f associés aux différents échantillons \tilde{m} se recouvrent, de manière à éviter des artefacts d'aliasing. Le jacobien de la transformation des pentes \tilde{m} en directions réfléchies ω_i est donné par :

$$\left\| \frac{\partial \omega_i}{\partial \tilde{n}} \right\| = \left\| \frac{\partial \omega_i}{\partial \omega_m} \right\| \left\| \frac{\partial \omega_m}{\partial \tilde{n}} \right\| = 4 |\omega_m \cdot \omega_o| |\omega_m \cdot \omega_g|^3.$$

Le calcul complet est donné en détail dans l'algorithme 3 et il est illustré dans la figure 10.2. Il produit des résultats avec une complexité constante et sans artefacts (mais possiblement biaisés).

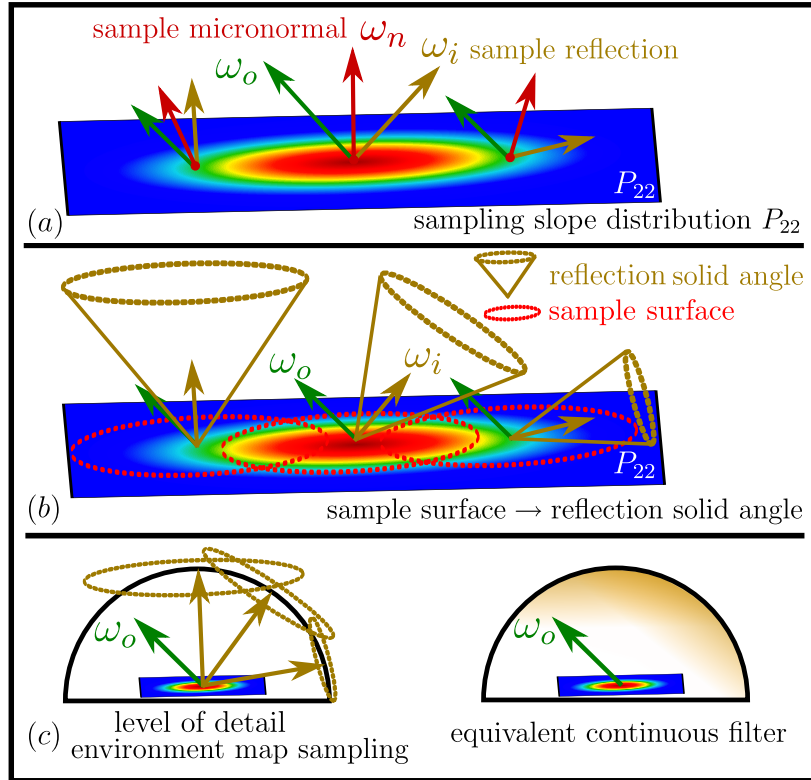


FIGURE 10.2 – Echantillonnage de l'éclairage environnemental pré-filtré. (a) Nous échantillons la distribution des pentes P^{22} . Pour chaque normale ω_m échantillonnée, nous calculons la direction réfléchie ω_i . (b) Nous transformons les surfaces recouvertes par les échantillons en espace des pentes par des angles solides dans l'espace des directions réfléchies en utilisant le jacobien de cette transformation. (c) Pour chaque direction réfléchie ω_i nous évaluons l'éclairage environnemental pré-filtré. Nous choisissons la taille de filtre en accédant à une cube map MIPmappée. La reconstruction est continue et sans artefacts.

Algorithm 3 Echantillonnage de l'éclairage environnemental pré-filtré

```

function ENVMAPSAMPLING( $\omega_o, \mathbb{E}[\tilde{m}], \sigma_x, \sigma_y, c_{xy}$ )
   $I = 0$ 
  for  $j, k = 1..N$  do
     $x_{\tilde{m}} = p_j \sigma_x + \mathbb{E}[x_{\tilde{m}}]$ 
     $y_{\tilde{m}} = (r_{xy} p_j + \sqrt{1 - r_{xy}^2} p_k) \sigma_y + \mathbb{E}[y_{\tilde{m}}]$ 
     $W_{\tilde{m}} = W_j W_k$ 
     $\omega_m = (-x_{\tilde{m}}, -y_{\tilde{m}}, 1) / \sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}$ 
     $\omega_i = 2 (\omega_m \cdot \omega_o) \omega_m - \omega_o$ 
     $J = 4 \times |\omega_m \cdot \omega_o| \times |\omega_m \cdot \omega_g|^3$ 
     $\alpha = J \times A$ 
     $I += W_{\tilde{m}} \times \frac{F(\omega_m, \omega_i) \frac{\langle \omega_m, \omega_o \rangle}{\omega_m \cdot \omega_g}}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} \times \text{textureLod}(\omega_i, \text{LOD}(\alpha))$ 
  end for
return  $I \times \frac{\omega_{\tilde{m}} \cdot \omega_g}{\omega_{\tilde{m}} \cdot \omega_o}$ 
end function

```

Discussion sur l'échantillonnage pré-filtré. Un des avantages de notre méthode est sa formulation dans l'espace des pentes dont la paramétrisation est cartésienne, contrairement à l'espace sphérique. En exploitant la structure de cette paramétrisation, nous pouvons échantillonner la BRDF de manière stable, même avec de petits nombres d'échantillons. En effet, la différence entre notre méthode d'échantillonnage pré-filtré et celle de *filtered importance sampling* [CK07, KC08] est que nous utilisons une grille régulière pour placer les échantillons tandis que cette technique précédente consiste à générer les échantillons par importance sampling. La figure 10.3 montre les filtres de reconstruction obtenus par les deux méthodes avec différents nombres d'échantillons. Nous observons que, bien que les deux méthodes convergent vers le même résultat, notre méthode est plus stable pour un faible nombre d'échantillons : le filtre de reconstruction a la bonne orientation moyenne lorsqu'il est calculé avec une grille régulière alors qu'il peut être décalé de manière importante lorsqu'il est calculé par importance sampling. C'est parce que générer les échantillons par importance sampling crée une reconstruction très aléatoire, ce qui crée des artefacts visibles lorsque le nombre d'échantillons est trop faible. Au contraire, le fait d'utiliser une grille régulière dans notre méthode introduit de la structure dans le schéma d'échantillonnage et le rend plus stable.

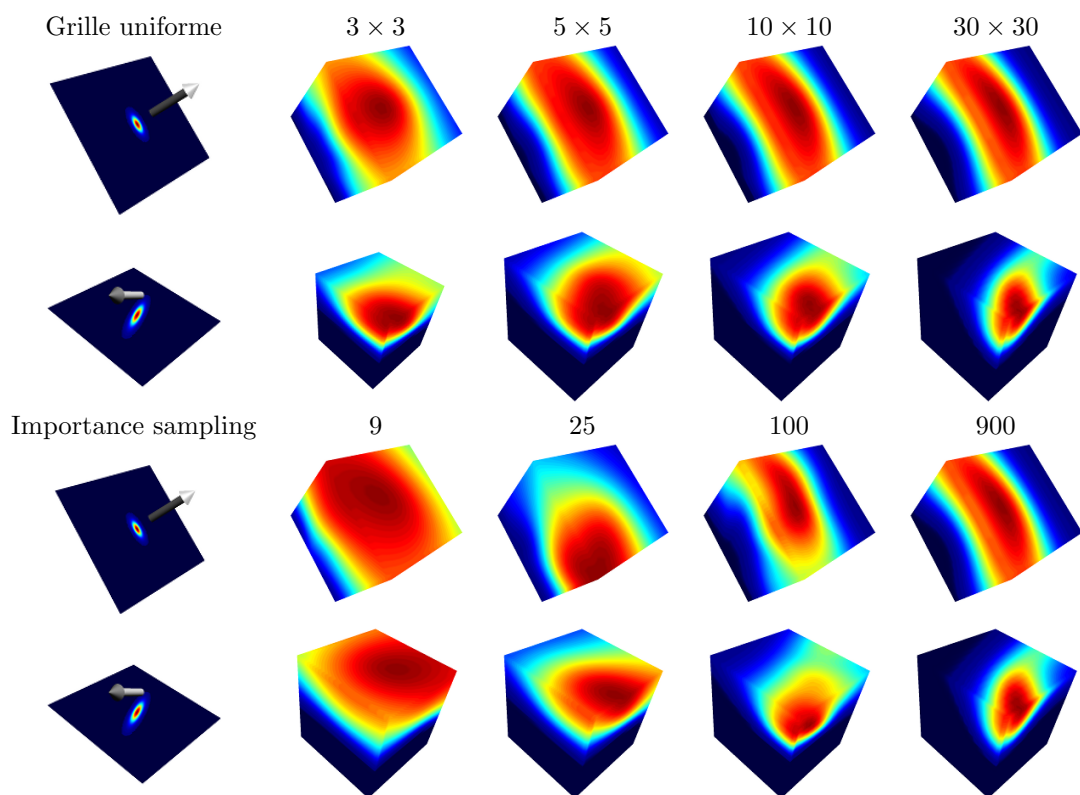


FIGURE 10.3 – Les filtres de reconstruction continus calculés dans des cube maps lorsque nous échantillonons avec (haut) une grille uniforme (notre méthode) (bas) de l'importance sampling. La colonne de gauche montre la direction du point de vue (la flèche) et la distribution P^{22} des pentes de la surface (le plan avec la gaussienne, sous la flèche).

10.3 Résultats

Rendus de validation. Pour valider notre modèle, nous le testons avec des données qui correspondent exactement à ses hypothèses : la figure 10.4 présente des résultats calculés sur des sphères dont la surface est enrichie avec une fonction de bruit aléatoire à statistiques gaussiennes. La ligne du haut présente les résultats de référence obtenus en utilisant la fonction de bruit pour générer une normal map, tandis que dans la ligne du milieu la fonction de bruit est utilisée pour générer une height map. Alors que les statistiques des distributions de normales sont les mêmes sur ces normal maps et ces height maps de référence, les apparences obtenues ne sont pas les mêmes : c’est parce que la visibilité n’intervient pas sur les normal maps où les micro-facettes n’existent pas géométriquement. Cela montre que même les méthodes les plus sophistiquées de filtrage de normal maps, comme [HSRG07], ne peuvent pas fonctionner pour capturer l’apparence des height maps, puisqu’elles ne peuvent pas faire mieux que capturer l’apparence de la référence pour les normal maps.

Dans la ligne du bas, le shading est calculé avec notre modèle de filtrage et de calcul. Ces résultats montrent qu’il capture bien l’apparence obtenue en calculant la référence sur la height map sur laquelle les effets de visibilité sont importants. De plus, lorsque le nombre N d’échantillons augmente, les résultats de notre modèle convergent exactement vers les résultats mesurés. Cela permet de valider deux choses : notre modèle est correct (les équations sont correctes) et notre schéma d’intégration numérique l’est aussi.

Concernant le choix de la taille de grille N pour l’intégration numérique, pour des micro-facettes diffuses, une grille de taille 5×5 restaure l’essentiel de l’apparence visuelle et l’erreur numérique est inférieure à 1% avec une grille de taille 9×9 . Pour obtenir le meilleur compromis qualité/performances, nous recommandons une taille de grille de 5×5 pour des micro-facettes diffuses.

Notre schéma d’échantillonnage pour des micro-facettes spéculaires avec de l’éclairage environnemental produit toujours des résultats sans artefacts visuels, mais dont le résultat peut être biaisé. Il faut un grand nombre d’échantillons pour réduire ce biais, en particulier pour les rugosités importantes ($\sigma > 0.25$). Dans les cas les plus difficiles, il faut utiliser une grille de taille 32×32 pour obtenir un biais inférieur à 1%. Le besoin de qualité dépend du type d’application et le compromis qualité/performances peut être ajusté en choisissant la valeur de N .

Rendu sur des données réelles. la figure 10.5 illustre d’autres comparaisons avec de la géométrie plus complexe. Le coussin (haut) est illuminé avec une irradiance map uniquement, et le serpent (bas) est éclairé avec une irradiance map et des sources ponctuelles.

Nous avons testé notre approche sur un modèle de production complexe et détaillé, voir figure 1.2, section 1.1.3. Ce résultat fait partie d’une animation, disponible dans la vidéo associée à l’article *LEADR Mapping* [DHI+13].

Nous appliquons également notre modèle sur des surfaces procédurales animées, comme celle de l’océan proposée par Bruneton et Neyret [BNH10]. Cela est illustré dans la figure 10.6. Nous utilisons notre formule analytique de la BRDF à micro-facettes spéculaires pour évaluer la contribution du soleil (qui est approximativement une source directionnelle), et notre schéma d’échantillonnage pré-filtré pour évaluer la contribution du ciel stockée dans une cube map.

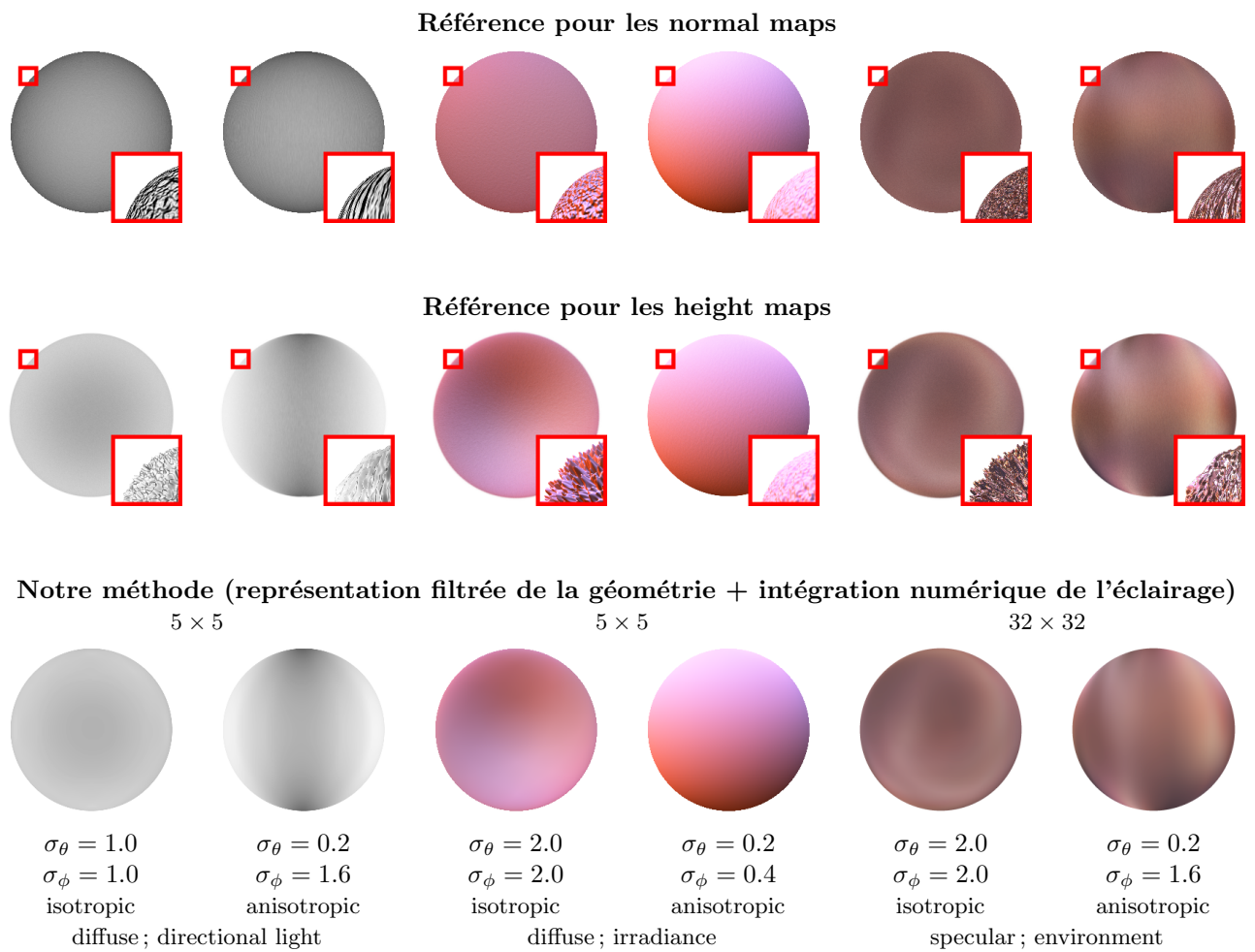


FIGURE 10.4 – Des images de référence générées avec (haut) des normal maps et (milieu) des height maps. (bas) Le shading calculé avec notre méthode d'intégration numérique avec des tailles de grille 5×5 pour des micro-facettes diffuses et 32×32 pour des micro-facettes spéculaires.

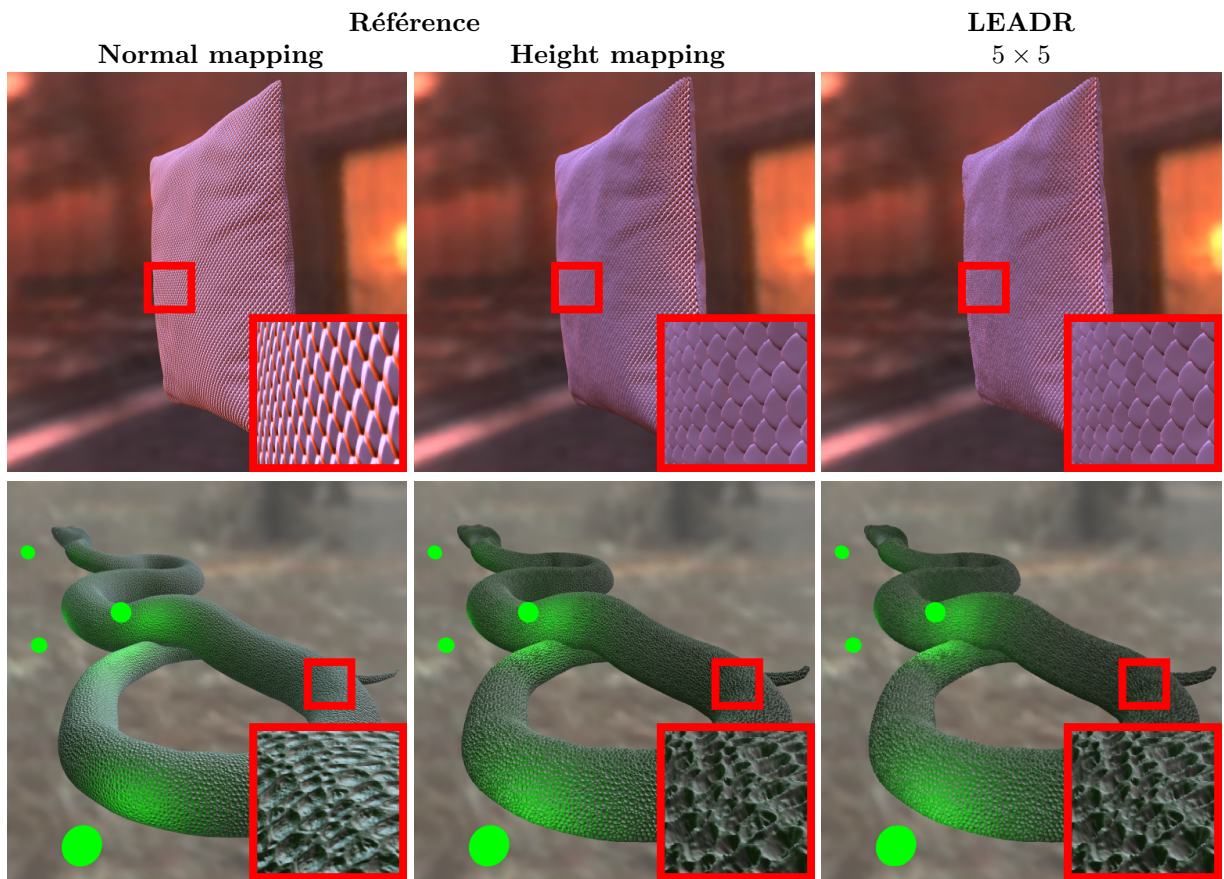


FIGURE 10.5 – Un éclairage environnemental appliqué sur des micro-facettes diffuses. (gauche) La référence pour le normal mapping ne capture pas les effets d'occlusion et de visibilité présents sur une height map (milieu). Ceux-ci sont présents dans notre modèle et intégrés correctement par notre méthode d'intégration numérique.

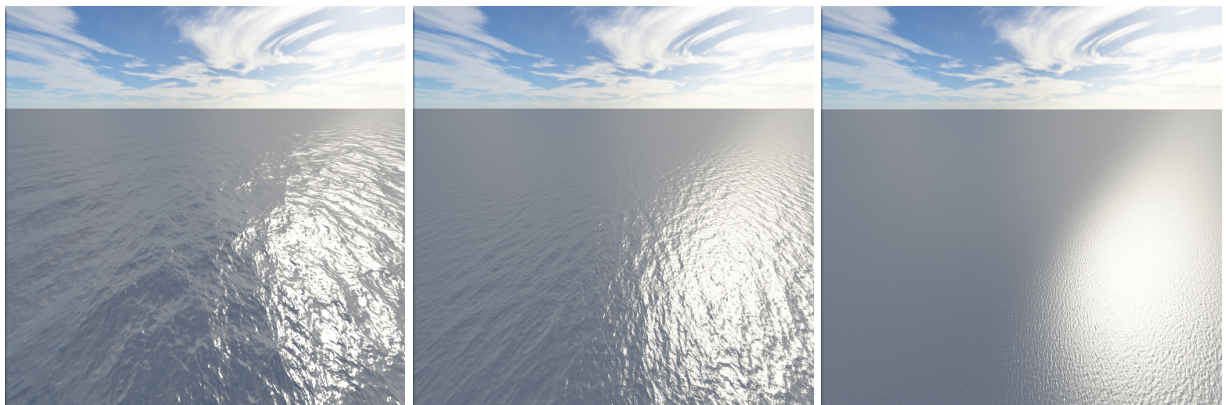


FIGURE 10.6 – Une surface d'océan multi-échelles procédurale et animée rendue avec notre technique. Le matériau de la surface est modélisé par des micro-facettes spéculaires dont la réflectance est progressivement filtrée lorsqu'on s'éloigne.

Performances. Nous avons mesuré les performances de notre algorithme sur un processeur Intel Core i7-930 CPU à 2.80 GHz et une carte graphique NVidia GTX 480. Pour mesurer les performances de nos algorithmes de shading uniquement, nous rendons un carré de taille 1024×1024 pour lequel nous shadons chaque pixel avec nos algorithmes. Les chiffres que nous reportons sont donnés en millisecondes.

Pour une source directionnelle, nous comparons l'évaluation analytique du shading de la partie spéculaire de la BRDF, l'évaluation de l'équation (10.1), avec les évaluations des BRDFs de Blinn-Phong [Bli77] et de LEAN Mapping [OB10] :

Constant	Blinn-Phong	LEAN	LEADR
0.055	0.146	0.266	0.267

Notre évaluation de la BRDF est aussi rapide que celui de LEAN Mapping bien qu'elle requière plus d'opérations. Cela est sans doute dû au fait que le goulot d'étranglement principal vient de l'accès aux textures.

Dans la table suivante, nous utilisons la même configuration, mais avec de l'éclairage environnemental, bien plus coûteux car il requiert une intégration numérique. Nous évaluons notre temps de shading pour différentes tailles de grilles d'échantillonnage. Les temps de calcul sont comparés à un seul accès texture dans la direction réfléchie moyenne :

Reflection	One sample	3×3	5×5	7×7
specular	0.402	0.433	1.704	4.284
diffuse	0.148	1.067	2.750	5.027

Comme illustré dans la figure 10.7, les performances de notre algorithme augmentent linéairement avec le nombre d'échantillons utilisés.

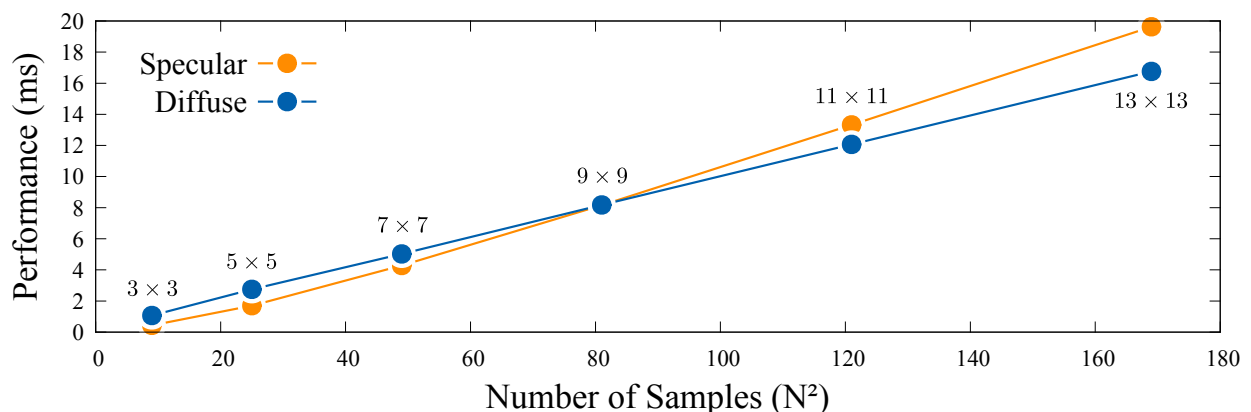


FIGURE 10.7 – Temps de calcul (ms) pour le rendu sur une NVidia GTX 480 avec notre méthode, affiché en fonction de la taille N de la grille d'échantillonnage.

Il serait certainement possible d'améliorer les performances en optimisant le placement des échantillons de manière à en utiliser le moins possible, par exemple avec la méthode de Jäckel [Jäc05]. Pour le modèle de production de la figure 1.2, notre algorithme a des performances entre 40 et 60 millisecondes par image pour une résolution de 1024×1024 . Pour les vues très proches ou très distantes (recouvrement du modèle d'environ $\approx 128 \times 128$ pixels sur l'écran), le temps de rendu est d'environ 200 et 9 millisecondes respectivement. A noter que dans la vidéo nous utilisons une carte graphique NVidia Quadro 6000 avec 6 GB de mémoire pour pouvoir rendre le modèle original avec tous ses détails, dont l'ensemble des textures fait 2.8 GB. Nous observons des performances similaires sur ce hardware.

De manière générale, notre méthode démontre une amélioration importante de la qualité visuelle du filtrage par rapport aux techniques de normal mapping, tout en maintenant des performances temps-réel. De plus, notre technique est temporellement et spatialement cohérente et produit toujours des images sans artefacts visuels.

10.4 Au-delà du modèle : attaquer les problèmes liés au filtrage de la géométrie

La limitation principale de notre modèle de filtrage des surfaces détaillées est qu’il ne s’applique que sur les détails des height maps stockées dans des textures. Notre modèle fonctionne très bien lorsque l’hypothèse sous laquelle la géométrie est localement plane est valide. Alors que notre modèle est parfaitement bien posé sous cette hypothèse, nous obtenons en pratique des images avec de nombreux problèmes si nous l’appliquons directement sur un modèle aussi détaillé que celui de la figure 1.2. En effet, sur une géométrie aussi complexe, nous arrivons vite dans le cas où certains pixels violent l’hypothèse de planarité du modèle et cela peut produire des artefacts qui rendent les images complètement inutilisables, même si cela se produit dans un faible nombre de pixels (en pratique un seul pixel qui clignote est extrêmement visible et dérangent et peut ruiner complètement une séquence animée).

Il faut bien comprendre que le modèle de filtrage que nous proposons n’est pas responsable de ces problèmes d’aliasing dus à la courbure géométrique. Le modèle n’est pas “incorrect” : en effet, il fonctionne très bien pour traiter ce pour quoi il a été créé : filtrer des height maps appliquées sur un voisinage parfaitement plane. Ces problèmes sont plutôt la responsabilité de ceux (nous) qui ont créé le modèle avec des hypothèses peu raisonnables ou de l’utilisateur (nous aussi) qui tente d’appliquer le modèle en dehors de son domaine de validité.

Dans cette section, nous discutons deux problèmes que nous avons dû résoudre pour calculer des images sans aliasing et utilisables en pratique.

10.4.1 Unifier le filtrage de la courbure avec celui de la height map

Problème. Lorsqu’une surface très spéculaire est courbe, la probabilité d’y observer un reflet est énorme, comme l’expliquent Takahashi et al. [SST89, TT91]. En effet, les courbures créent des distributions de normales très larges qui réfléchissent de la lumière depuis de nombreuses directions. On peut observer la présence de reflets spéculaires presque systématique sur les rebords courbes des objets de la vie courante, comme une tasse, un stylo, les touches d’un clavier, etc. Lorsqu’une courbure de ce genre se projette dans un seul pixel, la probabilité d’y observer un reflet est donc très importante. Mais notre modèle ne tient pas compte des distributions de normales sous-pixel introduites par la courbure de la géométrie macroscopique. Dans notre modèle, par défaut, la surface courbe est remplacée par une surface plate et la distribution de normales due à la courbure est remplacée par un dirac qui ne contient qu’une unique normale. Cette normale peut être responsable (ou pas) d’un reflet spéculaire vers une source de lumière selon la configuration de la scène. Dans ce cas, une infime variation de la position de la caméra ou de l’objet qui implique un changement de la valeur de cette normale peut faire passer subitement la quantité d’énergie réfléctée par la surface de 0 à une valeur énorme. La conséquence de cet effet est des pixels qui clignent fortement : c’est de l’aliasing spéculaire dû à la courbure.

Ces surfaces courbes introduisent donc de l’aliasing qui rend les images inutilisables en pratique, en particulier dans les séquences animées (des pixels qui clignent sont très visibles). La figure 10.8 montre, dans la colonne de gauche, des exemples de surfaces courbes qui produisent un aliasing très important.

Notre méthode. Pour rendre des images utilisables, nous utilisons une idée assez simple mais très efficace pour gérer l’aliasing qui vient de la courbure de la surface macroscopique. Notre idée, similaire à celle d’Amanatides [Ama92] et celle de Schilling [Sch01], est de convertir la distribution de normales sous-pixel générée par la courbure de la surface géométrique en une distribution de pentes que nous pouvons réinjecter dans notre modèle. Nous traitons alors, de manière unifiée, le filtrage de la courbure avec celui des height maps.

En pratique, dans le shader, nous utilisons les dérivées fournies par OpenGL pour la normale géométrique, $\frac{\partial \omega_g}{\partial X}$ et $\frac{\partial \omega_g}{\partial Y}$ pour calculer une déviation moyenne qui donne un estimateur de la courbure $\Delta \omega_g = \sqrt{\frac{\partial \omega_g}{\partial X} + \frac{\partial \omega_g}{\partial Y}}$. Pour une déviation de petite longueur, l’angle θ de la déviation correspond à : $\tan \theta = \Delta \omega_g$. Par ailleurs, nous avons la conversion normales/pentes donnée par : $\|\tilde{m}\| = \tan \theta$. De fait, lorsque la déviation est

raisonnablement petite, nous pouvons approximer une déviation d'amplitude $\Delta\omega_g$ par une pente d'amplitude $\|\tilde{m}\|$. Puisque nous souhaitons représenter l'ensemble des déviations des normales par une distribution de pentes, nous posons comme paramètre du rugosité : $\alpha = \Delta\omega_g$. Ce choix, à un facteur près, est arbitraire, car une distribution de Beckmann n'a pas d'amplitude maximale. Cela nous donne une distribution de pentes P_{curv}^{22} qui vient de la courbure. Pour l'introduire dans notre modèle, nous supposons que la courbure est indépendante du contenu des textures. Nous pouvons alors combiner la distribution de pentes P^{22} de la height map avec la distribution de pentes P_{curv}^{22} qui représente la courbure en les convoluant. Nous obtenons une distribution de pentes effectives :

$$P_{\text{eff}}^{22} = P^{22} * P_{\text{curv}}^{22},$$

que nous utilisons pour l'ensemble des calculs décrits dans ce chapitre. En pratique, cette convolution se calcule en ajoutant la variance α^2 de P_{curv}^{22} à celle de P^{22} . Cette technique, bien que peu précise, nous permet d'éliminer tous les artefacts visuels dans nos rendus et nous obtenons des images et des animations utilisables (voir figure 10.8, colonne de droite).



FIGURE 10.8 – (gauche) Notre modèle de filtrage qui ne tient compte que du contenu des height maps ne permet pas de filtrer les détails géométriques hautes-fréquences dus à la courbure de la géométrie du maillage. Cela produit de l'aliasing (les pixels blancs saturés), très visible par exemple sur les jointures du maillage entre les paupières et les yeux ou entre les griffes et les pattes. Pendant l'animation, ces pixels blancs saturés clignotent et, bien que peu nombreux, ruinent complètement la qualité de l'image. (droite) Notre méthode consiste à représenter la courbure comme une distribution de pentes que nous injectons dans celle de la height map. De cette manière, le shading de la courbure s'unifie avec celui des height maps et devient représentable et traitable par notre modèle. Avec cette technique nous anti-aliasons le shading dans les zones à courbure sous-pixel importante et nous obtenons des images anti-aliasées, propres et sans artefacts.

10.4.2 Résoudre les problèmes liés à la tessellation adaptative

Problème. Comme nous l'avons expliqué dans les chapitres 4 et 5, le concept d'aire projetée est important dans la définition d'une BRDF. En particulier, le résultat d'une BRDF est indéfini lorsque la valeur de l'aire projetée visible est négative. En théorie, cela n'arrive jamais car les normales auto-masquées ne sont pas visibles. Mais cela peut arriver en pratique lorsque l'échantillonnage de la tessellation de la surface n'est pas assez important par rapport à l'échantillonnage des textures. La différence de fréquence d'échantillonnage entre les triangles et les textures peut créer une légère divergence entre les normales effectives des triangles et les valeurs qu'elles sont supposées avoir dans la texture. A cause de cette divergence, des meso-normales supposées être invisibles peuvent être rendues et notre modèle est indéfini dans ces cas. Ce problème survient typiquement dans les silhouettes, lorsque les normales sont presque orthogonales à la direction d'observation, comme illustré dans la figure 10.9.

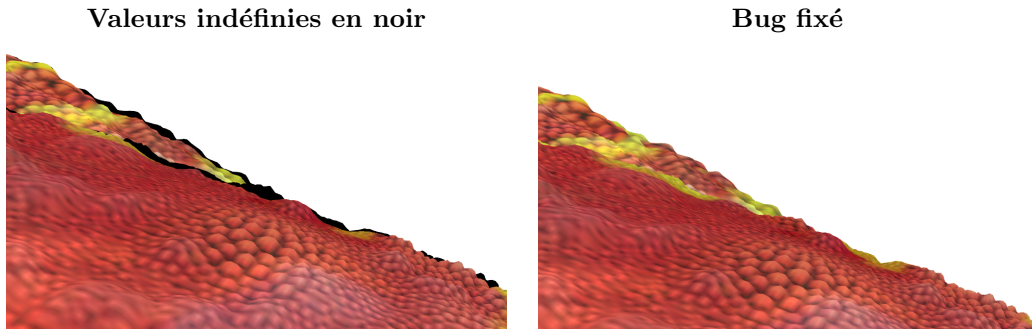


FIGURE 10.9 – (gauche) Certaines méso-normales qui devraient être auto-masquées peuvent être rendues dans les silhouettes lorsque la fréquence de tessellation des triangles est insuffisante par rapport à la fréquence d'échantillonnage des textures. Dans ces situations, notre algorithme retourne des valeurs indéfinies (pixels noirs). (droite) Nous résolvons ce problème en ajoutant une micro-normale artificielle toujours visible depuis la direction d'observation, de manière à ce que l'algorithme renvoie toujours une valeur calculable et bien définie.

Notre méthode. Nous résolvons ce problème dans nos schémas d'intégration numérique, décrits en section 10.2, en ajoutant une micro-normale artificielle $\tilde{\omega}_m$, calculée en biaisant la méso-normale vers la direction d'observation, assurant qu'elle soit toujours visible :

$$\tilde{\omega}_m = \frac{\omega_{\bar{m}} - (\omega_{\bar{m}} \cdot \omega_o) \epsilon \omega_o}{\|\omega_{\bar{m}} - (\omega_{\bar{m}} \cdot \omega_o) \epsilon \omega_o\|},$$

où nous posons $\epsilon = 1.01$ en pratique. La construction de la micro-normale artificielle $\tilde{\omega}_m$ est illustrée en figure 10.10. Nous utilisons $\tilde{\omega}_m$ dans nos algorithmes d'échantillonnage de la même manière que les autres micro-normales, mais avec un poids \tilde{W}_m très faible (nous posons $\tilde{W}_m = 0.00001$ en pratique).

Lorsque l'aire projetée des autres micro-normales est positive (ce qui est toujours supposé être le cas en théorie), la contribution de $\tilde{\omega}_m$ est négligeable à cause de son poids minuscule \tilde{W}_m . Lorsque toutes les autres micro-normales ont une aire projetée de 0 (ce qui ne devrait jamais être le cas en théorie), seule $\tilde{\omega}_m$ contribue à la valeur de l'intégrale. La transition entre ces deux cas est continue et le calcul de $\tilde{\omega}_m$ rend ce hack temporellement cohérent.

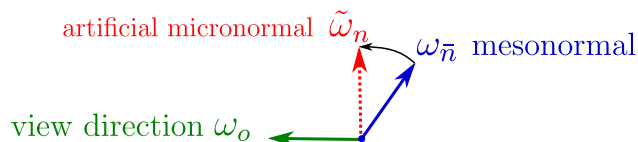


FIGURE 10.10 – Construction de la micro-normale artificielle.

10.5 Résumé du chapitre

Points clés du chapitre

- Utiliser une représentation multi-échelles pour les surfaces et les matériaux permet de transformer le problème d'intégration spatiale non scalable en un problème d'intégration statistique scalable.
- Pour obtenir l'apparence de la surface multi-échelles il faut néanmoins calculer ses interactions avec l'éclairage.
- Nous nous intéressons aux éclairages directionnels, ponctuels et de l'éclairage environnemental stocké dans des cube maps.
- L'évaluation de la BRDF est analytique avec des matériaux à micro-facettes spéculaires et une source d'éclairage ponctuelle ou directionnelle.
- Dans les autres cas, nous utilisons un schéma d'intégration numérique dont nous pouvons ajuster la précision et contrôler les performances.
- Pour éviter d'introduire de l'aliasing, nous combinons notre intégration numérique avec une représentation pré-filtrée de l'éclairage environnemental dans l'esprit de *filtered importance sampling*. Cela garantit une reconstruction sans aliasing, mais potentiellement biaisée avec un biais qui tend vers 0 lorsque le nombre d'échantillons augmente.
- Avec les modèles géométriquement très complexes, nous arrivons vite au moment où l'hypothèse de planarité locale est violée. Cela introduit de l'aliasing dans les images et les rend inutilisables.
- En pratique, nous utilisons pour la courbure géométrique sous-pixel la même représentation que pour la surface multi-échelles, ce qui nous permet de filtrer les deux en même temps et obtenir des images utilisables.

A voir

- L'annexe D propose des résultats de validation supplémentaires.
- L'article *LEADR Mapping* [DHI⁺13], son supplemental material et ses deux vidéos.

Chapitre 11

Importance sampling pour les BSDFs à micro-facettes

Dans le chapitre 10, nous avons proposé des algorithmes temps-réel pour intégrer l'éclairage avec la BRDF multi-échelles. Dans ce chapitre, nous proposons une manière de réaliser cette intégration mais avec un cahier des charges adapté aux exigences du rendu offline. Les moteurs de rendu offline sont typiquement basés sur des algorithmes de rendu de type Monte Carlo. Ceux-ci ont recours à des techniques d'*importance sampling* (échantillonnage préférentiel) pour évaluer efficacement l'éclairage et les matériaux. Dans le but de rendre notre modèle de BRDF multi-échelles compatible avec ce type de rendu, nous lui associons, dans ce chapitre, une technique d'importance sampling. Mais la technique que nous proposons ne se limite pas au modèle de BRDFs à micro-facettes à distribution gaussienne que nous avons introduit dans les chapitres 8 et 9. Notre technique s'applique à n'importe quel type de matériau à micro-facettes, y compris aux matériaux réfractifs modélisés par des BTDFs (Bidirectional Transmittance Distribution Functions). En conséquence, nous présentons ce chapitre sous un angle plus général, celui du rendu des BSDFs (Bidirectional Scattering Distribution Functions) à micro-facettes.

Bien que des techniques d'importance sampling existent déjà dans la littérature pour ces types de matériaux [WMLT07], l'originalité de la technique présentée dans ce chapitre est qu'elle tire parti des connaissances du modèle introduites dans le chapitre 4. Dans la section 11.2, nous verrons que la technique d'importance sampling précédente, la plus répandue et considérée état de l'art, ne tient pas compte de la manière dont le modèle est construit, ce qui la rend inefficace. En effet, cette technique est basée entièrement sur l'usage de la distribution de normales pour générer des micro-facettes aléatoirement. Mais nous avons vu que la quantité clé de la théorie des micro-facettes n'est pas la distribution de normales mais la distribution de normales *visibles*, où les normales sont pondérées par leur visibilité depuis la direction incidente. C'est à partir de cette distribution visible qu'est construit le modèle de BSDF à micro-facettes. Négliger la visibilité dans la technique d'importance sampling produit des échantillons avec des poids très importants, responsables des *fireflies artefacts* (les pixels très brillants) dans les figures 11.1 et 11.2. De plus, aux angles rasants, la moitié des micro-facettes générées sont auto-masquées et ont une contribution nulle, ce qui rend la technique inefficace. A la place, nous proposons dans la section 11.3 de générer des micro-facettes avec la distribution de normales visibles. De cette manière, beaucoup plus proche du modèle, les poids des échantillons ne contiennent que la fonction d'ombrage, et sont donc toujours à valeur dans $[0, 1]$: les valeurs arbitrairement élevées qui sont responsables des fireflies artefacts sont évitées et la variance est réduite. De plus, par construction, cette technique ne génère jamais de micro-facettes auto-masquées et évite de gaspiller l'espace d'échantillonnage. Pour le même budget de temps de calcul, nous obtenons des images avec moins d'artefacts et de bruit. Enfin, cette stratégie d'échantillonnage, basée sur les propriétés du modèle, dépend du profil de micro-surface. Nous décrivons sa mise en oeuvre pour les deux modèles présentés au chapitre 4 : le modèle à V-cavités (11.4) et le modèle de Smith (11.5).

Previous : Importance sampling de la BSDF avec la distribution de normales 64 spp (10.0s)

Our : Importance sampling de la BSDF avec la distribution de normales visibles 58 spp (10.4s)

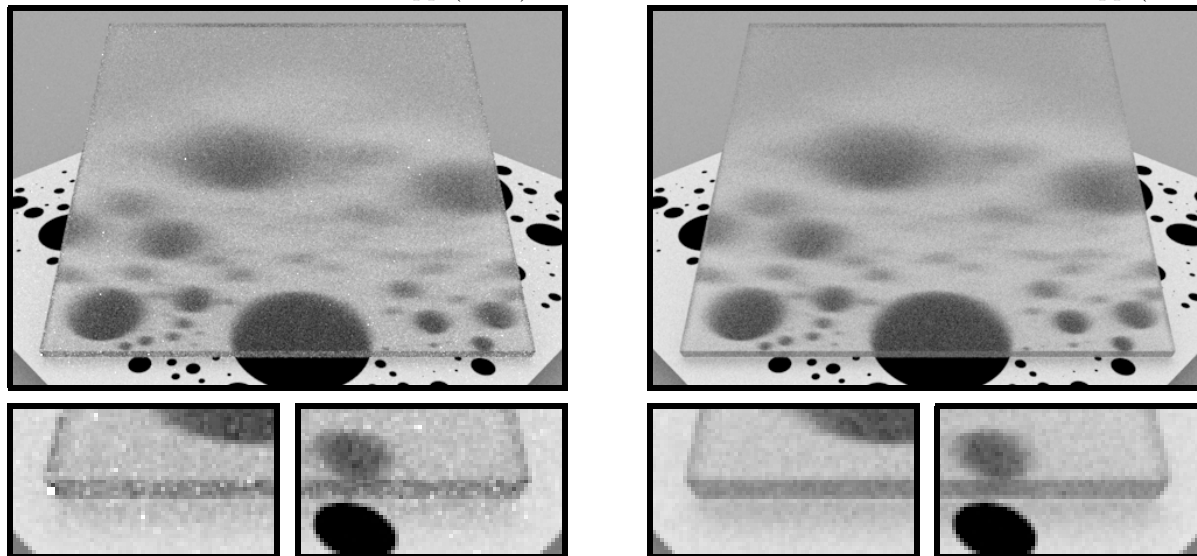


FIGURE 11.1 – Un plateau fait d'un matériau diélectrique ($n = 1.5$) avec une rugosité anisotrope (GGX $\alpha_x = 0.05$, $\alpha_y = 0.4$) dont le profil de micro-surface est celui de Smith. Pour le même budget temps d'échantillonnage, notre technique (à droite) réduit significativement la variance et converge plus vite que la technique standard (à gauche).



FIGURE 11.2 – Un objet fait d'un matériau conducteur avec une rugosité isotrope (GGX $\alpha = 0.05$) dont le profil de micro-surface est celui de Smith. (64 échantillons par pixel).

11.1 Le modèle de BSDF à micro-facettes

La théorie des micro-facettes que nous avons revisitée dans la partie I a été étendue au phénomène de transmission, modélisé par la BTDF, par les travaux de Stam [Sta01] et repris par Walter et al. [WMLT07]. Dans la section 4.3, nous avons vu que les modèles de BRDFs à micro-facettes sont dérivés à partir de la distribution de normales visibles. Intuitivement, la distribution des directions réfléchies donnée par la BRDF est modélisée à partir des réflexions spéculaires des micro-facettes visibles. Lorsque le matériau de la micro-surface est un diélectrique, une partie de l'énergie est réfléchi de cette manière (la BRDF) et une autre est transmise (la BTDF). De la même manière que la BRDF, la distribution des directions transmises données par la BTDF est modélisée à partir des transmissions spéculaires des micro-facettes visibles. C'est ce que montre la figure 11.3.

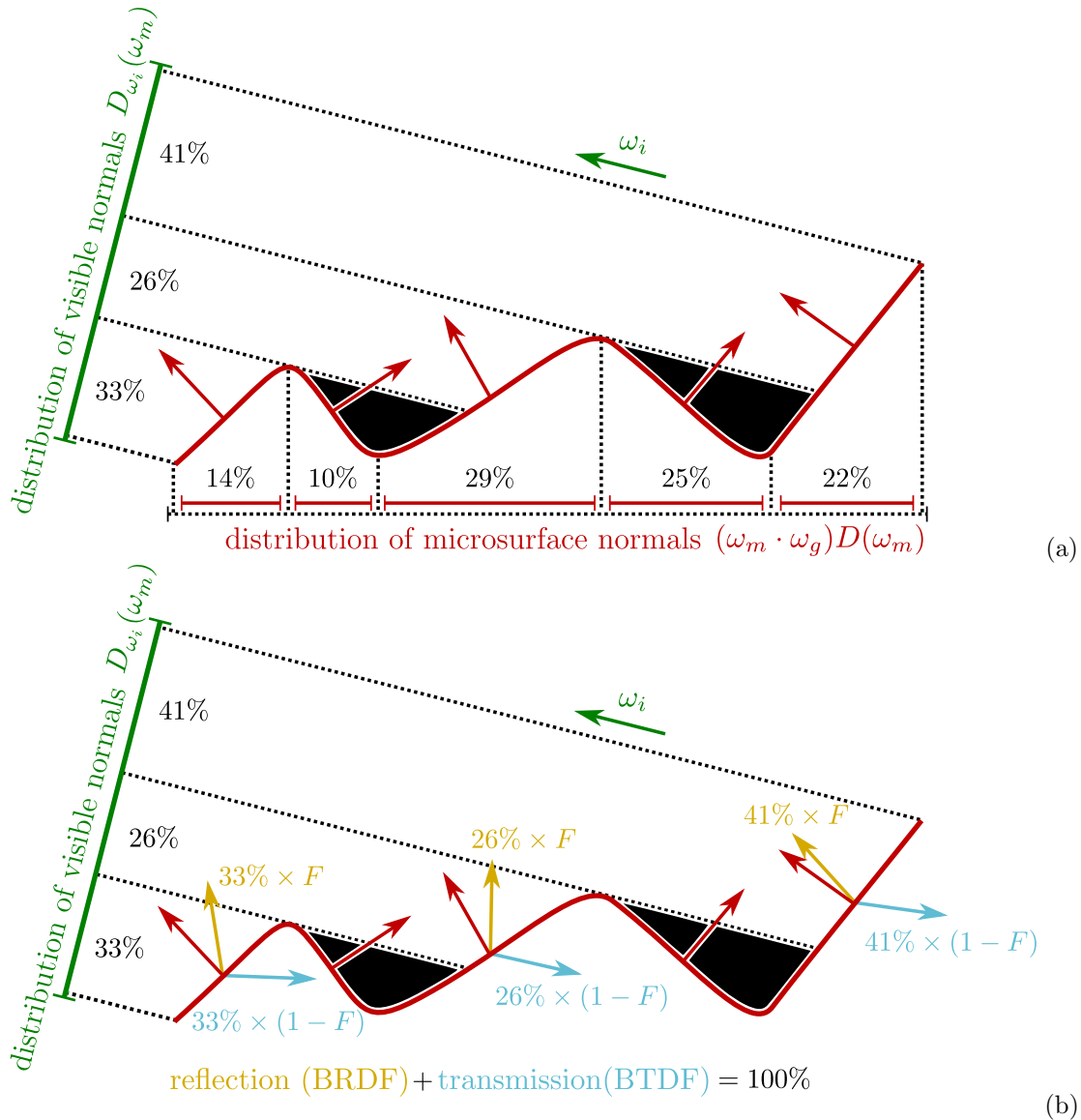


FIGURE 11.3 – Le modèle de BSDF à micro-facettes est dérivé en appliquant les opérateurs “reflect” et “transmit” sur la distribution des normales visibles. Les proportions de réflexions et de transmissions sont données par le terme de Fresnel F .

Dans le chapitre 4, nous avons redérivé l'ensemble des résultats de la théorie des micro-facettes s'appliquant aux BRDFs. Ci-dessous, nous rappelons leur généralisation aux BSDFs proposée par Walter et al. [WMLT07].

L'équation d'illumination locale. La BSDF f est utilisée dans l'équation d'illumination locale donnée par :

$$L(\omega_i) = \int_{\Omega} L(\omega_o) |\omega_o \cdot \omega_g| f(\omega_i, \omega_o) d\omega_o, \quad (11.1)$$

où $L(\omega_o)$ est la radiance dans la direction sortante. La BSDF $f(\omega_i, \omega_o) = f_r(\omega_i, \omega_o) + f_t(\omega_i, \omega_o)$ est la somme de la BRDF

$$f_r(\omega_i, \omega_o) = \frac{F(\omega_i, \omega_{hr}) G_2(\omega_i, \omega_o, \omega_{hr}) D(\omega_h)}{4 |\omega_i \cdot \omega_g| |\omega_o \cdot \omega_g|},$$

et de la BTDF

$$f_t(\omega_i, \omega_o) = \frac{|\omega_i \cdot \omega_{ht}| |\omega_o \cdot \omega_{ht}| n_0^2 (1 - F(\omega_i, \omega_{ht})) G_2(\omega_i, \omega_o, \omega_{ht}) D(\omega_{ht})}{|\omega_i \cdot \omega_g| |\omega_o \cdot \omega_g| (n_i(\omega_i \cdot \omega_{ht}) + n_o(\omega_o \cdot \omega_{ht}))^2},$$

où ω_{hr} et ω_{ht} sont les half vectors pour la réflexion et la transmission respectivement, et n_i et n_o sont les indices de réfraction pour les parties extérieures et intérieures du matériau respectivement.

La fonction de masquage et d'ombrage de la BTDF dans le modèle à V-cavités. Dans le modèle à V-cavités, la fonction de masquage et d'ombrage tient compte des corrélations sur les hauteurs dans la BRDF : ω_o est dans l'hémisphère supérieur ($\omega_o \cdot \omega_g > 0$) et le masquage et l'ombrage sont sur la même partie de la micro-facette et sont corrélés. C'est la raison pour laquelle la fonction d'ombrage et de masquage s'écrit comme le minimum de l'ombrage et du masquage.

Nous proposons de généraliser cela à la BTDF : ω_o est sur la partie inférieure de la micro-surface et le masquage et l'ombrage ont lieu sur les parties opposées. Cette anti-corrélation se modélise par $G_2(\omega_i, \omega_o, \omega_m) = \max(G_1(\omega_i, \omega_m) + G_1(\omega_o, \omega_m) - 1, 0)$ et la fonction de masquage et d'ombrage de la BSDF est donnée par :

$$G_2(\omega_i, \omega_o, \omega_m) = \begin{cases} \min(G_1(\omega_i, \omega_m), G_1(\omega_o, \omega_m)), & \text{si } \omega_o \cdot \omega_g > 0 \\ \max(G_1(\omega_i, \omega_m) + G_1(\omega_o, \omega_m) - 1, 0), & \text{sinon.} \end{cases}$$

La fonction de masquage et d'ombrage de la BTDF dans le modèle de Smith. Dans le modèle de Smith, la fonction de masquage et d'ombrage pour la partie transmise est obtenue simplement en inversant la direction de la normale géométrique ω_g , comme expliqué dans [WMLT07]. En effet, la rugosité de l'interface du matériau est la même vue de l'intérieur ou de l'extérieur. Lorsqu'on modélise la partie transmise dans l'hémisphère inférieure, il suffit de redéfinir l'hémisphère inférieure comme étant l'hémisphère supérieure pour sa ramener au cas précédent. Cela se fait en inversant la direction de ω_g .

11.2 La technique précédente d'importance sampling

Typiquement, dans les moteurs de rendu, des techniques d'importance sampling sont utilisées pour résoudre l'équation (11.1). La technique la plus courante d'importance sampling pour les BSDFs à micro-facettes utilise la distribution de normales pour générer les échantillons [War92, AS00, APS00, KSK01, Ash07, WMLT07]. Dans ce chapitre, nous l'appelons la méthode précédente et elle est dénotée par “*previous*” dans les figures. Elle est illustrée dans l'algorithme 4. Pour importance sampler le produit de la BSDF et du cosinus $|\omega_o \cdot \omega_g| f(\omega_i, \omega_o)$, l'algorithme commence par générer des normales ω_m avec la PDF $(\omega_m \cdot \omega_g) D(\omega_m)$ plutôt que des directions sortantes ω_o . Ensuite, un opérateur de transport de la lumière est appliqué sur la normale pour générer la direction sortante. L'opérateur *reflect*

$$\omega_o = 2|\omega_i \cdot \omega_m| \omega_m - \omega_i,$$

et *transmit*

$$\omega_o = \left(nc - \text{sign}(\omega_i \cdot \omega_g) \sqrt{1 + n(c^2 - 1)} \right) \omega_m - n\omega_i,$$

avec $c = (\omega_i \cdot \omega_m)$, et $n = \frac{n_i}{n_o}$,

sont choisis aléatoirement avec une probabilité qui dépend de la valeur du terme de Fresnel : $F(\omega_i, \omega_m)$ est la probabilité de réflexion et $1 - F(\omega_i, \omega_m)$ est la probabilité de transmission. La PDF résultante de cette procédure est la PDF, utilisée pour générer les normales, multipliée par le jacobien de l'opérateur de transport appliqué :

$$\left\| \frac{\partial \omega_{hr}}{\partial \omega_i} \right\| = \frac{1}{4|\omega_i \cdot \omega_{hr}|},$$

$$\left\| \frac{\partial \omega_{ht}}{\partial \omega_i} \right\| = \frac{n_o^2 |\omega_o \cdot \omega_{ht}|}{(n_i(\omega_i \cdot \omega_h) + n_o(\omega_o \cdot \omega_h))^2}.$$

Puisque ces jacobiens sont présents dans la PDF utilisée pour générer les échantillons et dans les expressions de la BRDF/BTDF, ils se simplifient dans l'expression du poids des échantillons. Par ailleurs, puisque la réflexion et la transmission sont choisies aléatoirement avec une probabilité qui dépend du terme de Fresnel, cette valeur, présente dans la BRDF/BSDF, est aussi présente dans la PDF d'échantillonnage et se simplifie dans l'expression du poids. Finalement, dans les deux cas le poids de l'échantillon vaut :

$$\text{weight}(\omega_o) = \frac{|\omega_m \cdot \omega_i| G_2(\omega_m, \omega_i, \omega_o)}{|\omega_i \cdot \omega_g| |\omega_m \cdot \omega_g|}.$$

Discussion. Cet algorithme d'échantillonnage est presque parfait en incidence frontale, mais on peut voir dans la figure 11.4 qu'il a deux problèmes majeurs aux angles d'incidence rasants.

Le premier problème est que la moitié des normales ω_m générées sont auto-masquées ($\omega_i \cdot \omega_m < 0$). Les poids associés valent 0 et les échantillons ne contribuent pas à l'évaluation de la BSDF : la moitié de l'espace d'échantillonnage est gâché.

Le second problème aux angles rasants est que les poids des échantillons peuvent être arbitrairement élevés et produisent une variance élevée (en revanche la valeur des poids est bornée à 2 avec le modèle des V-cavités). Ces valeurs arbitrairement élevées sont responsables des artefacts visibles dans la figure 11.2. Walter et al. [WMLT07] évoquent ce problème et proposent d'utiliser une BSDF avec une rugosité α' légèrement plus élevée aux angles rasants : $\alpha' = (1.2 - 0.2\sqrt{|\omega_i \cdot \omega_m|})\alpha$. Nous appelons ceci “Walter's trick” dans la figure 11.4. Le “Walter's trick” réduit significativement la valeur du poids maximal avec la distribution de Beckmann mais est peu efficace avec la distribution GGX. Par ailleurs, il ne résout pas le problème que la moitié des normales générées sont auto-masquées.

Algorithm 4 Sample ω_o (previous)

 $\omega_m \leftarrow \text{Simulate } (\omega_m \cdot \omega_g)D(\omega_m)$
if $\mathcal{U} < F(\omega_i, \omega_m)$ **then**
 $\omega_o \leftarrow \text{reflect}(\omega_i, \omega_m)$
else
 $\omega_o \leftarrow \text{transmit}(\omega_i, \omega_m)$
end if
 $\text{weight}(\omega_o) \leftarrow \frac{|\omega_m \cdot \omega_i| G_2(\omega_m, \omega_i, \omega_o)}{|\omega_i \cdot \omega_g| |\omega_m \cdot \omega_g|}$

Algorithm 5 Sample ω_o (our)

 $\omega_m \leftarrow \text{Simulate } D_{\omega_i}(\omega_m)$ \triangleright Algorithm 6 or 7
if $\mathcal{U} < F(\omega_i, \omega_m)$ **then**
 $\omega_o \leftarrow \text{reflect}(\omega_i, \omega_m)$
else
 $\omega_o \leftarrow \text{transmit}(\omega_i, \omega_m)$
end if
 $\text{weight}(\omega_o) \leftarrow \frac{G_2(\omega_m, \omega_i, \omega_o)}{G_1(\omega_i, \omega_m)}$

	previous	our
PDF(ω_m)	$(\omega_m \cdot \omega_g)D(\omega_m)$	$D_{\omega_i}(\omega_m)$
weight(ω_o)	$\frac{ \omega_m \cdot \omega_i G_2(\omega_m, \omega_i, \omega_o)}{ \omega_i \cdot \omega_g \omega_m \cdot \omega_g }$	$\frac{G_2(\omega_m, \omega_i, \omega_o)}{G_1(\omega_m, \omega_i)}$
V-cavities G	weight(ω_o) $\in [0, 2]$	weight(ω_o) $\in [0, 1]$
Smith G	weight(ω_o) $\in [0, \infty[$	weight(ω_o) $\in [0, 1]$

TABLE 11.1 – Propriétés des algorithmes d'importance sampling.

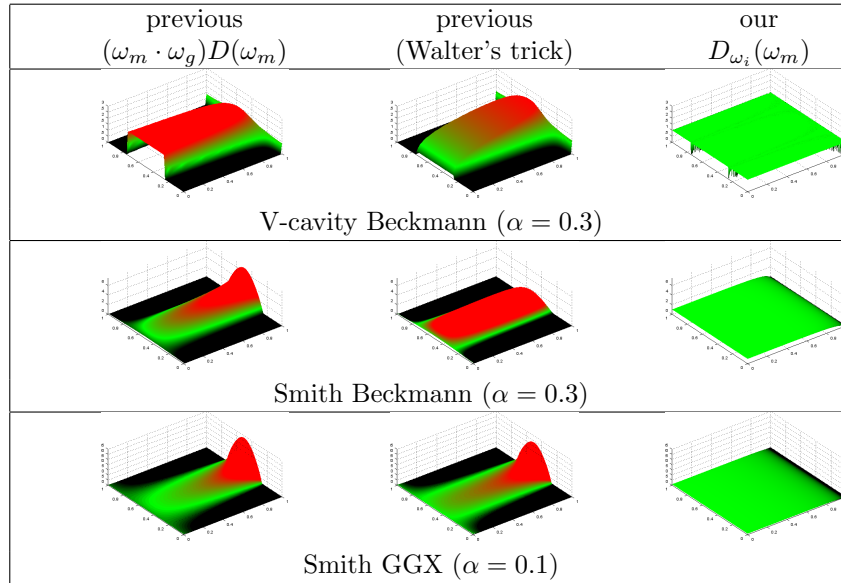


FIGURE 11.4 – Les poids des échantillons avec différentes PDFs en incidence rasante ($\theta_i = 1.5$). L'espace d'échantillonnage est paramétré par deux nombres aléatoires ($\mathcal{U}_1, \mathcal{U}_2$) utilisés pour générer ω_m . Notre technique d'importance sampling remplit l'espace d'échantillonnage avec des poids très proches de 1.0. (Le graphe pour notre méthode avec le modèle à V-cavités est bruité car nous utilisons un nombre aléatoire \mathcal{U}_3 supplémentaire, voir la section 11.4).

11.3 Notre technique d'importance sampling

Idée générale. L'idée principale que nous proposons est d'utiliser la distribution des normales visibles D_{ω_i} comme PDF d'échantillonnage, comme illustré dans l'algorithme 5. Les différences avec la méthode précédente sont affichées en rouge dans les algorithmes.

En faisant cela, nous simulons virtuellement ce qui est illustré dans la figure 11.3(b) : le rayon incident ω_i intersecte une normale aléatoire choisie parmi la distribution de normales visibles et ce rayon est ensuite réfléchi ou transmis. Seule la fonction d'ombrage doit être ajoutée, car elle n'émerge pas directement de ce modèle mais est présente pour forcer le modèle à ne représenter que le premier rebond sur la micro-surface, comme expliqué dans la section 4.3.

Propriétés. La valeur du poids des échantillons $\frac{G_2(\omega_m, \omega_i, \omega_g)}{G_1(\omega_m, \omega_i)}$ est la valeur de l'ombrage moyen visible. A noter que lorsque le masquage et l'ombrage ne sont pas corrélés ($G_2(\omega_i, \omega_o, \omega_m) = G_1(\omega_i, \omega_m)G_1(\omega_o, \omega_m)$), le poids est l'ombrage seul $G_2(\omega_i, \omega_o, \omega_m)/G_1(\omega_i, \omega_m) = G_1(\omega_o, \omega_m)$. Cela est intuitif, puisque si le masquage et l'ombrage ne sont pas corrélés, alors l'ombrage visible est simplement l'ombrage.

Comparaisons. La table 11.1 compare la méthode précédente avec la notre pour les profils de micro-surface à V-cavités et de Smith. Puisque $G_1 \leq G_2$, les poids sont toujours inférieurs à 1 dans notre méthode : les *fireflies artifacts* ne sont pas produits (voir figure 11.2). Cela est observable dans la figure 11.4. Nous voyons aussi que notre méthode ne gâche pas l'espace d'échantillonnage, contrairement à la méthode précédente. C'est parce que nous ne générons jamais de normales auto-masquées.

Implémentation. Les sections 11.4 et 11.5 sont dédiées à implémenter la méthode **Simulate** $D_{\omega_i}(\omega_m)$ de l'algorithme 5 pour des micro-surfaces avec les profils à V-cavités ou de Smith, respectivement.

11.4 Simuler $D_{\omega_i}(\omega_m)$ avec le modèle à V-cavités

11.4.1 Algorithme

La simulation de la PDF $D_{\omega_i}(\omega_m)$ avec le modèle à V-cavités est donnée par l'algorithme 6. Nous commençons par simuler la PDF des normales de la micro-surface $(\omega_m \cdot \omega_g) D(\omega_m)$ exactement comme dans la méthode précédente. Cela nous donne une micro-surface avec les normales duales ω_m et ω_m' (voir la figure 11.6). Ensuite, nous choisissons aléatoirement entre ω_m et ω_m' , où leur probabilité d'être choisies sont pondérées respectivement par $\langle \omega_i, \omega_m \rangle$ et $\langle \omega_i, \omega_m' \rangle$, qui sont leurs aires projetées dans la direction ω_i . Ce choix aléatoire pondéré par les aires projetées est ce qui transforme la PDF $(\omega_m \cdot \omega_g) D(\omega_m)$ en la PDF $D_{\omega_i}(\omega_m)$. En pratique, nous implémentons cela par un échange aléatoire de ω_m par ω_m' de probabilité $\frac{\langle \omega_i, \omega_m' \rangle}{\langle \omega_i, \omega_m \rangle + \langle \omega_i, \omega_m' \rangle}$.

Algorithm 6 Simulate $D_{\omega_i}(\omega_m)$ (V-cavity model)

$\omega_m \leftarrow \text{Simulate } (\omega_m \cdot \omega_g) D(\omega_m)$	▷ 1. choose microsurface
if $\mathcal{U} < \frac{\langle \omega_i, \omega_m' \rangle}{\langle \omega_i, \omega_m \rangle + \langle \omega_i, \omega_m' \rangle}$ then	▷ 2. choose normal
$\omega_m \leftarrow \omega_m'$	
end if	

La figure 11.5 illustre comment fonctionne l'algorithme 6. Lorsque le rayon incident approche de la surface géométrique, il commence par choisir une micro-surface à V-cavités de normales duales (ω_m, ω_m') . Puisque le modèle à V-cavités suppose que les micro-surfaces sont séparées, elles ont toutes la même visibilité et peuvent être choisies indépendamment de la direction incidente avec la PDF $(\omega_m \cdot \omega_g) D(\omega_m)$. Ensuite, une fois que la micro-surface est choisie, le rayon incident choisit une des deux normales ω_m ou ω_m' avec une probabilité pondérée par leur aires projetées. Nous incorporons ce second choix dans la PDF finale par un échange aléatoire entre les deux normales. Notre algorithme requiert donc un nombre aléatoire \mathcal{U} supplémentaire.

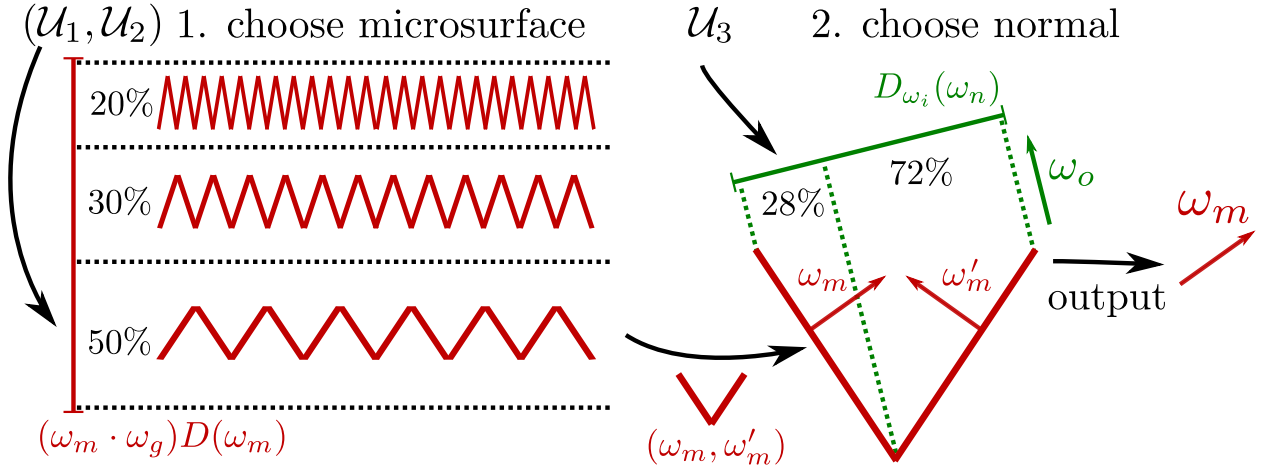


FIGURE 11.5 – Illustration de l'algorithme 6.

11.4.2 Preuve de l'algorithme

Pour vérifier que l'algorithme 6 simule effectivement la PDF $D_{\omega_i}(\omega_m)$, nous développons G_1 de l'équation (4.21) dans la définition de $D_{\omega_i}(\omega_m)$ de l'équation (4.6). L'opérateur min dans l'équation (4.21) est utilisé pour différencier deux cas, comme illustré dans la figure 11.6.

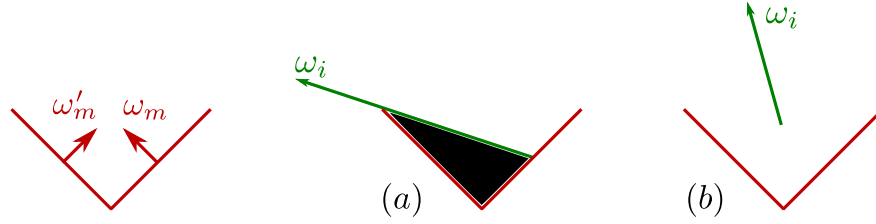


FIGURE 11.6 – Configuration de masquage avec le modèle à V-cavités.

Cas (a). Si ω_m' est auto-masquée, alors le masquage se produisant sur ω_m vaut $G_1(\omega_i, \omega_m) = 2 \frac{(\omega_m \cdot \omega_g)(\omega_i \cdot \omega_g)}{\langle \omega_i, \omega_m \rangle}$, et en utilisant le fait que $\langle \omega_i, \omega_m' \rangle = 0$ et $|\omega_i \cdot \omega_m| > 0$ on obtient :

$$\begin{aligned} D_{\omega_i}(\omega_m) &= 2 \frac{|\omega_m \cdot \omega_g| |\omega_i \cdot \omega_g| |\omega_i, \omega_m| D(\omega_m)}{|\omega_i, \omega_m| |\omega_i \cdot \omega_g|} \\ &= 2 \frac{|\omega_i \cdot \omega_m|}{|\omega_i \cdot \omega_m|} |\omega_m \cdot \omega_g| D(\omega_m) \\ &= 2 \frac{\langle \omega_i, \omega_m \rangle}{\langle \omega_i, \omega_m \rangle + 0} |\omega_m \cdot \omega_g| D(\omega_m) \\ &= 2 \frac{\langle \omega_i, \omega_m \rangle}{\langle \omega_i, \omega_m \rangle + \langle \omega_i, \omega_m' \rangle} (\omega_m \cdot \omega_g) D(\omega_m). \end{aligned}$$

Cas (b). Si ω_m et ω_m' sont toutes les deux non auto-masquées pour ω_i , alors il n'y a pas de masquage sur la micro-surface ($G_1 = 1$) et l'équation (4.6) devient :

$$\begin{aligned} D_{\omega_i}(\omega_m) &= \frac{\langle \omega_i, \omega_m \rangle D(\omega_m)}{|\omega_i \cdot \omega_g|} \\ &= 2 \frac{\langle \omega_i, \omega_m \rangle}{\langle \omega_i, \omega_m \rangle + \langle \omega_i, \omega_m' \rangle} (\omega_m \cdot \omega_g) D(\omega_m), \end{aligned}$$

$$\text{car } |\omega_i \cdot \omega_g| = \frac{\langle \omega_i, \omega_m \rangle + \langle \omega_i, \omega_m' \rangle}{2 \langle \omega_m, \omega_g \rangle}.$$

Conclusion. Dans les deux cas, la distribution des normales visibles peut s'écrire avec la même expression. Le facteur $\frac{\langle \omega_i, \omega_m \rangle}{\langle \omega_i, \omega_m \rangle + \langle \omega_i, \omega_m' \rangle}$ représente la proportion d'aire projetée entre ω_m et ω_m' . Dans l'algorithme 6, le remplacement aléatoire de ω_m par ω_m' multiplie la PDF $(\omega_m \cdot \omega_g) D(\omega_m)$ par le facteur $\frac{\langle \omega_i, \omega_m \rangle}{\langle \omega_i, \omega_m \rangle + \langle \omega_i, \omega_m' \rangle}$, et le facteur 2 est présent car ω_m peut être choisie pour deux micro-surfaces (celle associée à ω_m et celle associée à ω_m') et réciproquement pour ω_m' .

11.5 Simuler $D_{\omega_i}(\omega_m)$ avec le modèle de Smith

Nous avons vu en section 11.4 que simuler la PDF des normales visibles est relativement facile avec le modèle à V-cavités. Il n'en est pas de même avec le modèle de Smith qui est plus complexe. Pour y arriver, nous utilisons plusieurs changements d'espace et transformations qui rendent le problème plus simple. Dans la section 11.5.1 nous allons voir qu'on peut simuler la PDF des normales visibles D_{ω_i} en simulant la PDF des pentes visibles $P_{\omega_i}^{22}$. Ensuite, en 11.5.2, nous montrons que $P_{\omega_i}^{22}$ est invariant par étirement lorsque c'est le cas de P^{22} . En pratique, cela implique qu'une procédure connue pour simuler $P_{\omega_i}^{22}$ avec une rugosité fixée peut être utilisée avec n'importe quelle rugosité ou anisotropie de la même distribution. Si ces étapes intermédiaires semblent a priori complexes et difficiles, nous verrons que la technique est, au final, relativement simple. L'algorithme complet est donné et discuté dans la section 11.5.4.

11.5.1 Formulation dans l'espace des pentes

Dans l'équation (4.19) nous avons défini D_{ω_i} pour le modèle de Smith. Cette équation s'écrit dans l'espace des pentes :

$$\begin{aligned} D_{\omega_i}(\omega_m, 1, 1) &= \frac{\langle \omega_i, \omega_m \rangle D(\omega_m)}{\int_{\Omega} \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m} \\ &= \frac{1}{c} \langle \omega_i, \omega_m \rangle D(\omega_m), \end{aligned} \quad (11.2)$$

où $c = \int_{\Omega} \langle \omega_i, \omega_m \rangle D(\omega_m) d\omega_m$ est le coefficient de normalisation de cette distribution. Nous rappelons que $\tilde{m} = (x_{\tilde{m}}, y_{\tilde{m}}) = (-\frac{x_n}{z_n}, -\frac{y_n}{z_n})$ est le vecteur de pente associé à la normale $\omega_m = (x_n, y_n, z_n)$ et réciproquement $\omega_m = \frac{(-x_{\tilde{m}}, -y_{\tilde{m}}, 1)}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}}$. Si la micro-surface est une height map, la distribution de normales est liée à la distribution des pentes $P^{22}(x_{\tilde{m}}, y_{\tilde{m}})$ par

$$(\omega_m \cdot \omega_g) D(\omega_m) = \left\| \frac{\partial \tilde{m}}{\omega_m} \right\| P^{22}(\tilde{m} = (x_{\tilde{m}}, y_{\tilde{m}})),$$

où $\left\| \frac{\partial \tilde{m}}{\omega_m} \right\| = \frac{1}{(\omega_m \cdot \omega_g)^3}$ est le jacobien de la transformation pentes/normales. Les distributions des normales visibles et des pentes visibles sont liées de la même manière par

$$D_{\omega_i}(\omega_m) = \left\| \frac{\partial \tilde{m}}{\partial \omega_m} \right\| P_{\omega_i}^{22}(\tilde{m} = (x_{\tilde{m}}, y_{\tilde{m}})). \quad (11.3)$$

En espace des pentes, l'équation (11.2) s'écrit

$$\begin{aligned} P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}) &= \\ \frac{1}{c} \chi^+(x_i x_{\tilde{m}} + y_i y_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + y_i y_{\tilde{m}} + z_i) &P^{22}(x_{\tilde{m}}, y_{\tilde{m}}), \end{aligned} \quad (11.4)$$

où le coefficient de normalisation en espace des pentes est

$$c = \int \int \chi^+(x_i x_{\tilde{m}} + y_i y_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + y_i y_{\tilde{m}} + z_i) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dx_{\tilde{m}} dy_{\tilde{m}}.$$

Première idée de l'algorithme. Si nous simulons $P_{\omega_i}^{22}$ et transformons la pente obtenue en une normale ω_m , la PDF simulée est D_{ω_i} . De cette manière, nous pouvons réaliser l'échantillonnage dans l'espace des pentes qui a une paramétrisation cartésienne et de nombreuses bonnes propriétés (comme nous allons le voir dans les sections suivantes). C'est la première idée de l'algorithme 7.

11.5.2 Invariance d'étirement de la distribution de pentes visibles

L'étirement de la micro-surface est illustré dans la figure 11.7. Les pentes de la micro-surface sont divisées par le coefficient d'étirement : $(x_{\tilde{m}}, y_{\tilde{m}})$ devient $(\frac{x_{\tilde{m}}}{\lambda_x}, \frac{y_{\tilde{m}}}{\lambda_y})$. L'opérateur d'étirement $\mathcal{S}^{\lambda_x, \lambda_y}(-)$, qui transforme la direction incidente normalisée $\omega_i = (x_i, y_i, z_i)$ en une autre direction incidente normalisée¹ est défini par

$$\mathcal{S}^{\lambda_x, \lambda_y}(\omega_i) = \frac{(\lambda_x x_i, \lambda_y y_i, z_i)}{\sqrt{(\lambda_x x_i)^2 + (\lambda_y y_i)^2 + z_i^2}}.$$

En section 4.5, nous avons défini les distributions de pentes invariantes par étirement (équation (4.32)) et avons montré que si P^{22} est invariante de forme, alors la fonction de masquage est invariante par étirement. Nous introduisons ici une propriété plus forte : *si la distribution des pentes P^{22} est invariante de forme, alors la distribution des pentes visibles $P_{\omega_i}^{22}$ est aussi invariante de forme.* Ceci s'écrit :

$$P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y) = \frac{1}{\lambda_x \lambda_y} P_{\mathcal{S}^{\lambda_x, \lambda_y}(\omega_i)}^{22}\left(\frac{x_{\tilde{m}}}{\lambda_x}, \frac{y_{\tilde{m}}}{\lambda_y}, \frac{\alpha_x}{\lambda_x}, \frac{\alpha_y}{\lambda_y}\right), \quad (11.5)$$

et est illustré dans la figure 11.7.

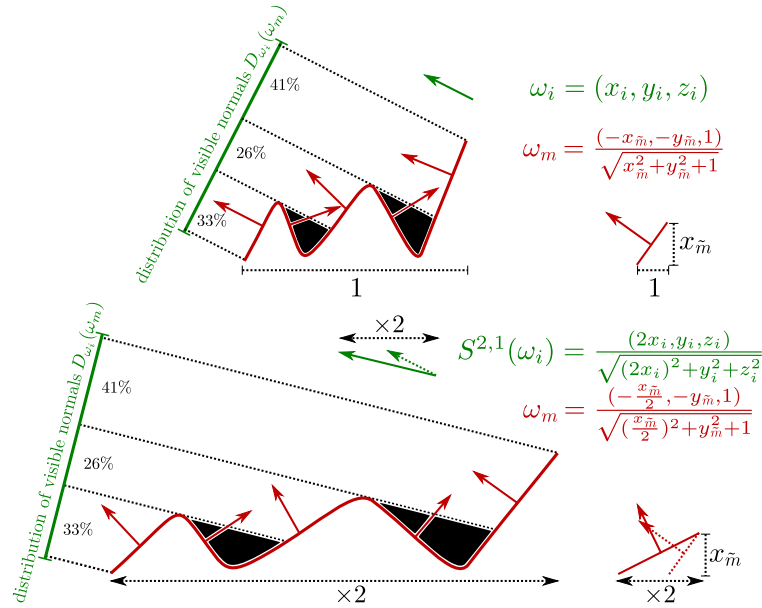


FIGURE 11.7 – Invariance par étirement de la distribution de pentes visibles.

1. A noter que les normales de la micro-surface ne sont pas modifiées de la même manière que la direction incidente, car ce sont des covecteurs. Les normales après l'étirement sont données par les pentes de la micro-surface étirée : $\omega_m = \frac{(-\frac{x_{\tilde{m}}}{\lambda_x}, -\frac{y_{\tilde{m}}}{\lambda_y}, 1)}{\sqrt{(\frac{x_{\tilde{m}}}{\lambda_x})^2 + (\frac{y_{\tilde{m}}}{\lambda_y})^2 + 1}}$.

Se ramener à une configuration isotrope. Une conséquence pratique de l'équation (11.5) est que la distribution des pentes visibles peut toujours s'exprimer avec une rugosité isotrope α (nous choisissons arbitrairement $\alpha = 1$) :

$$P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y) = \frac{1}{\alpha_x \alpha_y} P_{S^{\alpha_x, \alpha_y}(\omega_i)}^{22}\left(\frac{x_{\tilde{m}}}{\alpha_x}, \frac{y_{\tilde{m}}}{\alpha_y}, 1, 1\right). \quad (11.6)$$

Le facteur $\frac{1}{\alpha_x \alpha_y}$ dans la partie droite de l'équation est le jacobien de la transformation de $\left(\frac{x_{\tilde{m}}}{\alpha_x}, \frac{y_{\tilde{m}}}{\alpha_y}\right)$ en $(x_{\tilde{m}}, y_{\tilde{m}})$. Cette transformation est illustrée dans la figure 11.8.

Deuxième idée de l'algorithme. En pratique, si un algorithme de simulation est disponible pour $\left(\frac{x_{\tilde{m}}}{\alpha_x}, \frac{y_{\tilde{m}}}{\alpha_y}\right)$ avec la PDF $P_{S^{\alpha_x, \alpha_y}(\omega_i)}^{22}\left(\frac{x_{\tilde{m}}}{\alpha_x}, \frac{y_{\tilde{m}}}{\alpha_y}, 1, 1\right)$, nous pouvons l'utiliser pour simuler $(x_{\tilde{m}}, y_{\tilde{m}})$ avec la PDF $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y)$. De cette manière, nous n'avons pas à nous préoccuper des paramètres de rugosité α_x et α_y pour implémenter l'algorithme d'importance sampling. C'est la deuxième idée de l'algorithme 7.

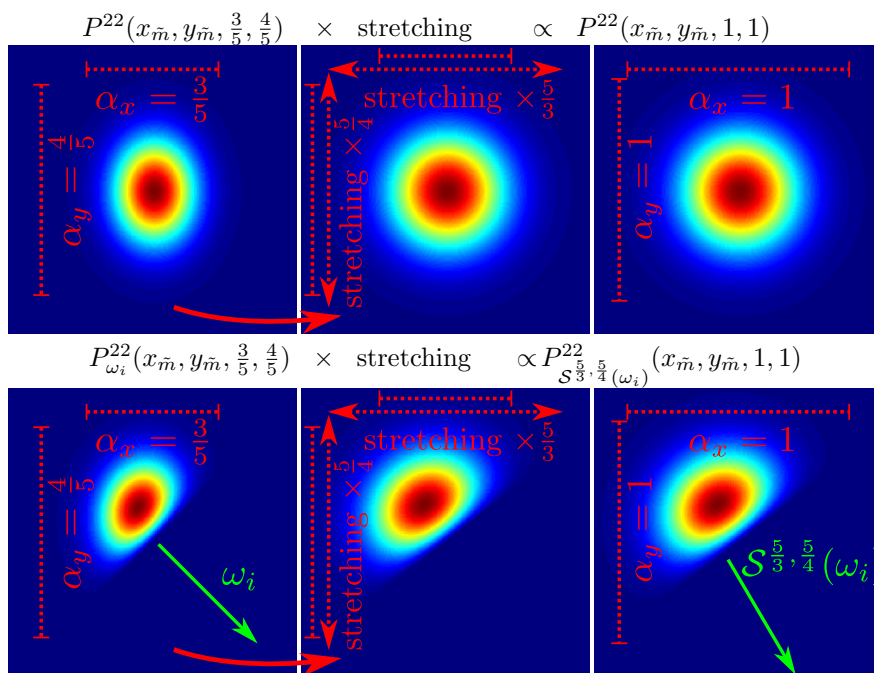


FIGURE 11.8 – Invariance par étirement de P^{22} et $P_{\omega_i}^{22}$ dans l'espace des pentes (exemple avec une distribution de Beckmann).

Preuve de l'équation (11.5) (invariance par étirement). Nous commençons par écrire l'équation (11.4). Puis, nous développons l'équation (4.32) et multiplions le numérateur et le dénominateur par $\frac{1}{\sqrt{(\lambda_x x_i)^2 + (\alpha_y y_i)^2 + z_i^2}}$. Enfin, au dénominateur, nous faisons le changement de variable : $\frac{1}{\lambda_x \lambda_y} dx_{\tilde{m}} dy_{\tilde{m}} = d\frac{x_{\tilde{m}}}{\lambda_x} d\frac{y_{\tilde{m}}}{\lambda_y}$.

$$\begin{aligned}
& P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y) \\
&= \frac{\chi^+(-x_i x_{\tilde{m}} - y_i y_{\tilde{m}} + z_i)(-x_i x_{\tilde{m}} - y_i y_{\tilde{m}} + z_i) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y)}{\int \int \chi^+(-x_i x_{\tilde{m}} - y_i y_{\tilde{m}} + z_i)(-x_i x_{\tilde{m}} - y_i y_{\tilde{m}} + z_i) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y) dx_{\tilde{m}} dy_{\tilde{m}}} \\
&= \frac{\frac{\chi^+(-\lambda_x x_i \frac{x_{\tilde{m}}}{\alpha_x} - \lambda_y y_i \frac{y_{\tilde{m}}}{\lambda_y} + z_i)(-\lambda_x x_i \frac{x_{\tilde{m}}}{\lambda_x} - \lambda_y y_i \frac{y_{\tilde{m}}}{\lambda_y} + z_i)}{\sqrt{(\lambda_x x_i)^2 + (\alpha_y y_i)^2 + z_i^2}} \frac{1}{\lambda_x \lambda_y} P^{22}\left(\frac{x_{\tilde{m}}}{\lambda_x}, \frac{y_{\tilde{m}}}{\lambda_y}, \frac{\alpha_x}{\lambda_x}, \frac{\alpha_y}{\lambda_y}\right)}{\int \int \frac{\chi^+(-\lambda_x x_i \frac{x_{\tilde{m}}}{\lambda_x} - \lambda_y y_i \frac{y_{\tilde{m}}}{\lambda_y} + z_i)(-\lambda_x x_i \frac{x_{\tilde{m}}}{\alpha_x} - \lambda_y y_i \frac{y_{\tilde{m}}}{\lambda_y} + z_i)}{\sqrt{(\lambda_x x_i)^2 + (\lambda_y y_i)^2 + z_i^2}} \frac{1}{\lambda_x \lambda_y} P^{22}\left(\frac{x_{\tilde{m}}}{\lambda_x}, \frac{y_{\tilde{m}}}{\lambda_y}, \frac{\alpha_x}{\lambda_y}, \frac{\alpha_y}{\lambda_y}\right) dx_{\tilde{m}} dy_{\tilde{m}}} \\
&= \frac{\frac{\chi^+(-\lambda_x x_i \frac{x_{\tilde{m}}}{\alpha_x} - \lambda_y y_i \frac{y_{\tilde{m}}}{\lambda_y} + z_i)(-\lambda_x x_i \frac{x_{\tilde{m}}}{\lambda_x} - \lambda_y y_i \frac{y_{\tilde{m}}}{\lambda_y} + z_i)}{\sqrt{(\lambda_x x_i)^2 + (\alpha_y y_i)^2 + z_i^2}} \frac{1}{\lambda_x \lambda_y} P^{22}\left(\frac{x_{\tilde{m}}}{\lambda_x}, \frac{y_{\tilde{m}}}{\lambda_y}, \frac{\alpha_x}{\lambda_y}, \frac{\alpha_y}{\lambda_y}\right)}{\int \int \frac{\chi^+(-\lambda_x x_i \frac{x_{\tilde{m}}}{\lambda_x} - \lambda_y y_i \frac{y_{\tilde{m}}}{\lambda_y} + z_i)(-\lambda_x x_i \frac{x_{\tilde{m}}}{\alpha_x} - \lambda_y y_i \frac{y_{\tilde{m}}}{\lambda_y} + z_i)}{\sqrt{(\lambda_x x_i)^2 + (\lambda_y y_i)^2 + z_i^2}} P^{22}\left(\frac{y_{\tilde{m}}}{\lambda_y}, \frac{y_{\tilde{m}}}{\lambda_y}, \frac{\alpha_x}{\lambda_y}, \frac{\alpha_y}{\lambda_y}\right) d\frac{x_{\tilde{m}}}{\lambda_x} d\frac{y_{\tilde{m}}}{\lambda_y}} \\
&= \frac{1}{\lambda_x \lambda_y} P_{S^{\lambda_x, \lambda_y}(\omega_i)}^{22}\left(\frac{x_{\tilde{m}}}{\lambda_x}, \frac{y_{\tilde{m}}}{\lambda_y}, \frac{\alpha_x}{\lambda_x}, \frac{\alpha_y}{\lambda_y}\right).
\end{aligned}$$

Invariance par étirement des hauteurs de Becker et Max. A noter qu'une idée similaire fut proposée par Becker et Max [BM93] : ils montrent que la distribution des normales visibles est invariante par étirement des hauteurs. Cela équivaut à étirer l'amplitude des pentes $\|\tilde{m}\|$, où les coordonnées des pentes $x_{\tilde{m}}$ et $y_{\tilde{m}}$ sont étirées par le même facteur. Cela permet de changer la rugosité en restant isotrope (ou avec le même facteur d'anisotropie). En revanche, nous exprimons l'opérateur d'étirement sur les composantes des pentes de manière séparable, ce qui permet de changer le ratio d'anisotropie.

11.5.3 Simuler $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$

Après avoir étiré la configuration, la distribution des pentes visibles est $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$. Pour simuler cette distribution, nous commençons par supposer, sans perte de généralité, que le repère local est tel que $\omega_i = (x_i, 0, z_i)$. Cela est possible puisque la distribution est isotrope ($\alpha_x, \alpha_y = 1$). L'expression de l'équation (11.4) se simplifie en

$$P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1) = \frac{1}{c} \chi^+(x_i x_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + z_i) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}).$$

Cette expression montre que le facteur de projection $\chi^+(x_i x_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + z_i)$ ne dépend pas de la composante $y_{\tilde{m}}$ du vecteur des pentes. Nous utilisons cette propriété pour simuler $P_{\omega_i}^{22}$ en deux étapes a. et b.

a. Simuler $P_{\omega_i}^{2-}(x_{\tilde{m}}, 1, 1)$ Nous commençons par simuler la variable aléatoire marginale $x_{\tilde{m}}$ donnée par la PDF :

$$\begin{aligned} P_{\omega_i}^{2-}(x_{\tilde{m}}, 1, 1) &= \frac{1}{c} \int_{-\infty}^{+\infty} \chi^+(x_i x_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + z_i) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1) dy_{\tilde{m}} \\ &= \frac{1}{c} \chi^+(x_i x_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + z_i) \int_{-\infty}^{+\infty} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1) dy_{\tilde{m}} \\ &= \frac{1}{c} \chi^+(x_i x_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + z_i) P^{2-}(x_{\tilde{m}}, 1, 1), \end{aligned} \quad (11.7)$$

où $P^{2-}(x_{\tilde{m}}, 1, 1)$ est la PDF marginalisée de $x_{\tilde{m}}$. Nous inversons

$$\begin{aligned} \mathcal{U}_1 &= \int_{-\infty}^{x_{\tilde{m}}} P_{\omega_i}^{2-}(x'_{\tilde{m}}, 1, 1) dx'_{\tilde{m}} \\ \Rightarrow x_{\tilde{m}} &= C_{\omega_i}^{2-^{-1}}(\mathcal{U}_1, 1, 1), \end{aligned} \quad (11.8)$$

où $C_{\omega_i}^{2-}$ est la CDF (Cumulative Distribution Function) associée à la PDF marginalisée $P_{\omega_i}^{2-}$.

Cette équation peut être résolue analytiquement pour les distributions de Beckmann et GGX (voir l'annexe F). Pour les autres distributions, si une solution analytique n'est pas dérivable, nous proposons de pré-calculer les solutions et de les stocker dans une table 2D table $x_{\tilde{m}} = T_{x_{\tilde{m}}}[\theta_i, \mathcal{U}_1]$. En effet, $x_{\tilde{m}}$ dépend seulement de deux variables : θ_i (car $x_i = \sin \theta_i$ et $z_i = \cos \theta_i$) et \mathcal{U}_1 .

b. Simuler $P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}, 1, 1)$ Ensuite, nous simulons la variable aléatoire conditionnelle $y_{\tilde{m}}|x_{\tilde{m}}$ donnée par la PDF

$$\begin{aligned} P_{\omega_i}^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}, 1, 1) &= \frac{\frac{1}{c} \chi^+(x_i x_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + z_i) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)}{\int_{-\infty}^{+\infty} \frac{1}{c} \chi^+(x_i x_{\tilde{m}} + z_i) (x_i x_{\tilde{m}} + z_i) P^{22}(x_{\tilde{m}}, y'_{\tilde{m}}, 1, 1) dy'_{\tilde{m}}} \\ &= \frac{P^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)}{\int_{-\infty}^{+\infty} P^{22}(x_{\tilde{m}}, y'_{\tilde{m}}, 1, 1) dy'_{\tilde{m}}} = P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}, 1, 1). \end{aligned} \quad (11.9)$$

Cela montre que la simulation de $y_{\tilde{m}}$ ne dépend pas de la direction incidente ω_i . De fait, nous inversons

$$\begin{aligned} \mathcal{U}_2 &= \int_{-\infty}^{y_{\tilde{m}}} P^{2|2}(y'_{\tilde{m}}|x_{\tilde{m}}, 1, 1) dy'_{\tilde{m}} \\ \Rightarrow y_{\tilde{m}} &= C^{2|2^{-1}}(\mathcal{U}_2|x_{\tilde{m}}, 1, 1), \end{aligned} \quad (11.10)$$

où $C^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}, 1, 1)$ est la CDF associée à la PDF $P^{2|2}$ qui est indépendante du point de vue.

Cette équation peut être résolue analytiquement pour les distributions de Beckmann et GGX (voir annexe F). Pour les autres distributions, si une solution analytique n'est pas dérivable, nous proposons de pré-calculer les solutions et de les stocker dans une table 2D $y_{\tilde{m}} = T_{y_{\tilde{m}}}[x_{\tilde{m}}, \mathcal{U}_2]$. En effet, $y_{\tilde{m}}|x_{\tilde{m}}$ dépend seulement de deux variables : $x_{\tilde{m}}$ and \mathcal{U}_2 .

11.5.4 Simuler $D_{\omega_i}(\omega_m)$

Dans l'algorithme 7, illustré en figure 11.9, nous donnons la technique complète pour simuler la PDF des normales visibles $D_{\omega_i}(\omega_m)$ avec le profil de Smith. Nous commençons par simuler $(x_{\tilde{m}}, y_{\tilde{m}})$ avec la PDF $P_{\mathcal{S}^{\alpha_x, \alpha_y}(\omega_i)}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$ en simulant $x_{\tilde{m}}$ avec la PDF marginalisée $P_{\omega_i}^{2-}(x_{\tilde{m}}, 1, 1)$ et $y_{\tilde{m}}$ avec la PDF conditionnelle $P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}, 1, 1)$. Ensuite, nous replaçons ce vecteur de pente dans le repère canonique où $\omega_i = (\cos \phi_i \sin \theta_i, \sin \phi_i \sin \theta_i, \cos \theta_i)$ en lui appliquant une rotation d'angle ϕ_i et nous inversons l'opération d'étirement en multipliant ses coordonnées par respectivement α_x et α_y . La PDF simulée à ce point est $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \alpha_x, \alpha_y)$. Finalement, nous transformons le vecteur de pente en une normale. La PDF simulée est $D_{\omega_i}(\omega_m, \alpha_x, \alpha_y)$.

Algorithm 7 Simulate $D_{\omega_i}(\omega_m)$ (Smith profile)

$\omega'_i \leftarrow \mathcal{S}^{\alpha_x, \alpha_y}(\omega_i)$	▷ 1. stretch
$x_{\tilde{m}} \leftarrow C_{\omega'_i}^{2--1}(\mathcal{U}_1, 1, 1)$	▷ 2.a. simulate $P_{\omega'_i}^{2-}(x_{\tilde{m}}, 1, 1)$
$y_{\tilde{m}} \leftarrow C^{2 2-1}(\mathcal{U}_2 x_{\tilde{m}}, 1, 1)$	▷ 2.b. simulate $P^{2 2}(y_{\tilde{m}} x_{\tilde{m}}, 1, 1)$
$\begin{pmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{pmatrix} \leftarrow \begin{pmatrix} \cos \phi'_i & -\sin \phi'_i \\ \sin \phi'_i & \cos \phi'_i \end{pmatrix} \begin{pmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{pmatrix}$	▷ 3. rotate
$\begin{pmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{pmatrix} \leftarrow \begin{pmatrix} \alpha_x x_{\tilde{m}} \\ \alpha_y y_{\tilde{m}} \end{pmatrix}$	▷ 4. unstretch
$\omega_m \leftarrow \frac{(-x_{\tilde{m}}, -y_{\tilde{m}}, 1)}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}}$	▷ 5. compute normal

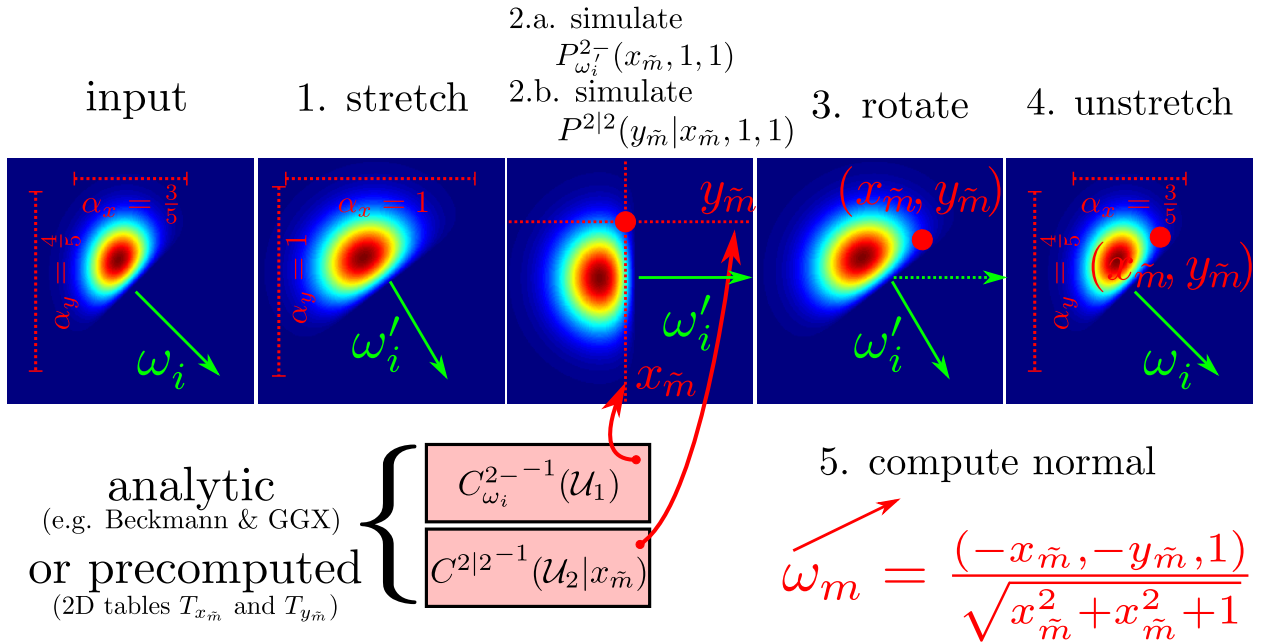


FIGURE 11.9 – Illustration of Algorithm 7.

Calcul analytique de $C_{\omega_i}^{2-}$ et $C^{2|2-}$. La pierre angulaire de l’algorithme 7 est l’inversion des CDFs $C_{\omega_i}^{2-}$ et $C^{2|2}$. Nous montrons comment réaliser cette inversion de manière analytique dans l’annexe F pour les distributions de Beckmann et GGX.

Précalcul de $C_{\omega_i}^{2-}$ et $C^{2|2-}$. Si une procédure d’inversion analytique des CDFs $C_{\omega_i}^{2-}$ et $C^{2|2}$ n’est pas disponible, nous proposons d’utiliser des données pré-calculées $T_{x_{\tilde{m}}}[\theta_i, \mathcal{U}_1]$ et $T_{y_{\tilde{m}}}[x_{\tilde{m}}, \mathcal{U}_2]$ paramétrées en espace des pentes. Un des avantages de l’espace des pentes est que l’équation de $P_{\omega_i}^{2|2}$ devient séparable : $\chi^+(x_i x_{\tilde{m}} + z_i)(x_i x_{\tilde{m}} + z_i)$ ne dépend pas de y_i et peut donc être sorti de l’intégrale. Cette propriété est utilisée dans les équations (11.7) et (11.9) et rend les pré-calculs plus simples qu’ils ne le seraient s’ils étaient exprimés en coordonnées sphériques. Le problème d’utilisation des coordonnées sphériques est discuté par Becker et Max [BM93] : ils stockent des tables paramétrées en (θ, ϕ) et font l’hypothèse que θ et ϕ peuvent être échantillonnés séparément, ce qui est approximatif et biaise l’échantillonnage, ou alors ils doivent utiliser une table 3D pour échantillonner θ pour un ϕ donné (ou réciproquement). Cela requiert une table 3D énorme et rendrait la technique inutilisable (≈ 1 GB). En revanche, dans l’espace des pentes, l’équation (11.9) montre que $P_{\omega_i}^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}, 1, 1) = P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}, 1, 1)$ et l’inversion ne dépend plus de θ_i , ce qui permet d’utiliser une table 2D indexée par $x_{\tilde{m}}$ et \mathcal{U}_2 . De fait, nous n’avons besoin d’utiliser que deux tables 2D $T_{x_{\tilde{m}}}[\theta_i, \mathcal{U}_1]$ et $T_{y_{\tilde{m}}}[x_{\tilde{m}}, \mathcal{U}_2]$. De plus, grâce à la propriété d’invariance par étirement en espace des pentes, nous pouvons utiliser les mêmes données avec différentes rugosités et anisotropies. Dans notre implémentation, $T_{x_{\tilde{m}}}$ et $T_{y_{\tilde{m}}}$ sont tous les deux de taille 1024^2 . Les distributions les plus populaires, comme GGX, ont à la fois des pics spéculaires très forts, mais aussi des queues longues (on parle de distributions à queues lourdes en statistiques), et donc requièrent une table de résolution 1024^2 pour les capturer précisément sur un domaine assez large. Avec ces deux tables, l’empreinte mémoire totale est alors de 32MB pour une classe entière de micro-surface, à rugosité et anisotropie variable.

Limites. A noter que cette technique ne fonctionne que lorsque la distribution de normales D est construite à partir d’une distribution de pentes invariante par étirement $P^{2|2}$. C’est le cas, par exemple, des distributions de Beckmann et GGX, mais ce n’est pas le cas de la distribution de Phong (mais une forme close et utilisable de la fonction de masquage de Smith n’existe pas pour la distribution de Phong de toute façon [WMLT07].)

11.5.5 Simuler $D_{\omega_i}(\omega_m)$ avec le modèle de Smith décentré

Dans le chapitre 5, nous avons étendu la théorie des micro-facettes aux distributions décentrées, condition indispensable pour la mise au point d'un modèle multi-échelles basé sur cette théorie. Nous allons voir que notre algorithme d'importance sampling pour le modèle de Smith s'adapte aussi aux distributions décentrées, et il est par conséquent utilisable avec le modèle multi-échelles du chapitre 5.

Invariance de torsion de la distribution de pentes visibles. Nous avons vu que la fonction de masquage est invariante par torsion. Nous allons maintenant montrer qu'il en va de même pour la distribution des pentes visibles. L'opération de torsion, illustrée en figure 11.10, est définie par :

$$\mathcal{T}^{\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}}(\omega_i) = \frac{(x_i, y_i, z_i - \bar{x}_{\tilde{m}}x_i - \bar{y}_{\tilde{m}}y_i)}{\sqrt{x_i^2 + y_i^2 + (z_i - \bar{x}_{\tilde{m}}x_i - \bar{y}_{\tilde{m}}y_i)^2}},$$

et l'invariance de torsion de la distribution de pentes visibles s'écrit :

$$P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}, \alpha_x, \alpha_y) = P_{\mathcal{T}^{\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}}(\omega_i)}^{22}(x_{\tilde{m}} - \bar{x}_{\tilde{m}}, y_{\tilde{m}} - \bar{y}_{\tilde{m}}, 0, 0, \alpha_x, \alpha_y). \quad (11.11)$$

En conséquence, nous pouvons utiliser la même idée que celle de la section précédente, c'est-à-dire se ramener à une configuration où une stratégie d'échantillonnage est connue, pour des configurations décentrées

$$\begin{aligned} P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, \bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}, \alpha_x, \alpha_y) &= P_{\mathcal{T}^{\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}}(\omega_i)}^{22}(x_{\tilde{m}} - \bar{x}_{\tilde{m}}, y_{\tilde{m}} - \bar{y}_{\tilde{m}}, 0, 0, \alpha_x, \alpha_y) \\ &= \frac{1}{\alpha_x \alpha_y} P_{S^{\alpha_x, \alpha_y}(\mathcal{T}^{\bar{x}_{\tilde{m}}, \bar{y}_{\tilde{m}}}(\omega_i))}^{22}\left(\frac{x_{\tilde{m}} - \bar{x}_{\tilde{m}}}{\alpha_x}, \frac{y_{\tilde{m}} - \bar{y}_{\tilde{m}}}{\alpha_y}, 0, 0, 1, 1\right). \end{aligned}$$

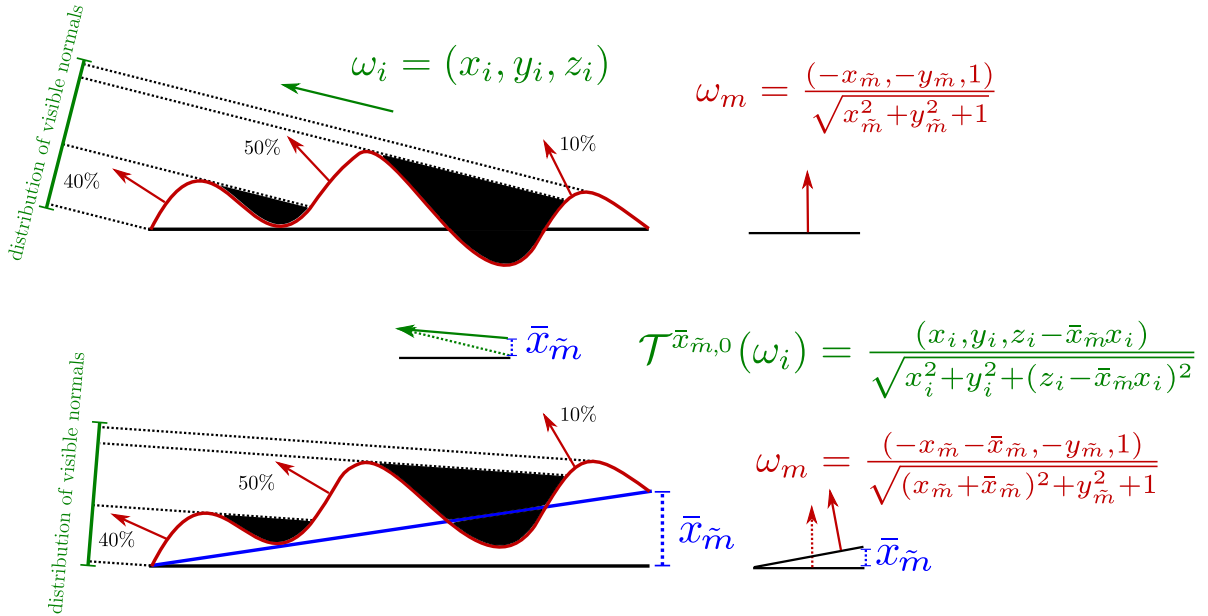


FIGURE 11.10 – Invariance par torsion de la distribution de pentes visibles.

Preuve de l'équation (11.11) (invariance par torsion).

$$\begin{aligned}
& P_{\omega_i}^{22}(x_{\bar{m}}, y_{\bar{m}}, \bar{x}_{\bar{m}}, \bar{y}_{\bar{m}}, \alpha_x, \alpha_y) \\
&= \frac{\chi^+(-x_i x_{\bar{m}} - y_i y_{\bar{m}} + z_i)(-x_i x_{\bar{m}} - y_i y_{\bar{m}} + z_i) P^{22}(x_{\bar{m}}, y_{\bar{m}}, \bar{x}_{\bar{m}}, \bar{y}_{\bar{m}}, \alpha_x, \alpha_y)}{\int \int \chi^+(-x_i x_{\bar{m}} - y_i y_{\bar{m}} + z_i)(-x_i x_{\bar{m}} - y_i y_{\bar{m}} + z_i) P^{22}(x_{\bar{m}}, y_{\bar{m}}, \bar{x}_{\bar{m}}, \bar{y}_{\bar{m}}, \alpha_x, \alpha_y) dx_{\bar{m}} dy_{\bar{m}}} \\
&= \frac{\chi^+(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}} + \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}} + \bar{y}_{\bar{m}}) + z_i)(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}} + \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}} + \bar{y}_{\bar{m}}) + z_i) P^{22}(x_{\bar{m}} - \bar{x}_{\bar{m}}, y_{\bar{m}} - \bar{y}_{\bar{m}}, 0, 0, \alpha_x, \alpha_y)}{\int \int \chi^+(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}} + \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}} + \bar{y}_{\bar{m}}) + z_i)(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}} + \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}} + \bar{y}_{\bar{m}}) + z_i) P^{22}(x_{\bar{m}} - \bar{x}_{\bar{m}}, y_{\bar{m}} - \bar{y}_{\bar{m}}, 0, 0, \alpha_x, \alpha_y) dx_{\bar{m}} dy_{\bar{m}}} \\
&= \frac{\chi^+(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}}) + z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}})(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}}) + z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}}) P^{22}(x_{\bar{m}} - \bar{x}_{\bar{m}}, y_{\bar{m}} - \bar{y}_{\bar{m}}, 0, 0, \alpha_x, \alpha_y)}{\int \int \chi^+(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}}) + z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}})(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}}) + z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}}) P^{22}(x_{\bar{m}} - \bar{x}_{\bar{m}}, y_{\bar{m}} - \bar{y}_{\bar{m}}, 0, 0, \alpha_x, \alpha_y) dx_{\bar{m}} dy_{\bar{m}}} \\
&= \frac{\sqrt{x_i^2 + y_i^2 + (z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}})^2}}{\sqrt{x_i^2 + y_i^2 + (z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}})^2}} \times \\
&= \frac{\chi^+(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}}) + z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}})(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}}) + z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}}) P^{22}(x_{\bar{m}} - \bar{x}_{\bar{m}}, y_{\bar{m}} - \bar{y}_{\bar{m}}, 0, 0, \alpha_x, \alpha_y)}{\int \int \chi^+(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}}) + z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}})(-x_i(x_{\bar{m}} - \bar{x}_{\bar{m}}) - y_i(y_{\bar{m}} - \bar{y}_{\bar{m}}) + z_i - x_i \bar{x}_{\bar{m}} - y_i \bar{y}_{\bar{m}}) P^{22}(x_{\bar{m}} - \bar{x}_{\bar{m}}, y_{\bar{m}} - \bar{y}_{\bar{m}}, 0, 0, \alpha_x, \alpha_y) dx_{\bar{m}} dy_{\bar{m}}} \\
&= P_{\mathcal{T}_{\bar{x}_{\bar{m}}, \bar{y}_{\bar{m}}}(\omega_i)}^{22}(x_{\bar{m}} - \bar{x}_{\bar{m}}, y_{\bar{m}} - \bar{y}_{\bar{m}}, 0, 0, \alpha_x, \alpha_y)
\end{aligned}$$

Idee pour simuler les distributions décentrées. Si un algorithme de simulation est disponible pour $P_{\omega_i}^{22}(x_{\bar{m}}, y_{\bar{m}}, 0, 0, 1, 1)$, alors nous pouvons l'utiliser pour simuler $P_{\omega_i}^{22}(x_{\bar{m}}, y_{\bar{m}}, \bar{x}_{\bar{m}}, \bar{y}_{\bar{m}}, \alpha_x, \alpha_y)$. En utilisant cette propriété, nous transformons l'algorithme 7 en l'algorithme 8.

Algorithm 8 Simulate $D_{\omega_i}(\omega_m)$ (Smith profile)

$$\begin{aligned}
\omega_i' &\leftarrow \mathcal{S}^{\alpha_x, \alpha_y}(\mathcal{T}_{\bar{x}_{\bar{m}}, \bar{y}_{\bar{m}}}(\omega_i)) && \triangleright \text{1. shear and stretch} \\
x_{\bar{m}} &\leftarrow C_{\omega_i'}^{2-1}(\mathcal{U}_1, 1, 1) && \triangleright \text{2.a. simulate } P_{\omega_i'}^{2-}(x_{\bar{m}}, 1, 1) \\
y_{\bar{m}} &\leftarrow C^{2|2-1}(\mathcal{U}_2 | x_{\bar{m}}, 1, 1) && \triangleright \text{2.b. simulate } P^{2|2}(y_{\bar{m}} | x_{\bar{m}}, 1, 1) \\
\begin{pmatrix} x_{\bar{m}} \\ y_{\bar{m}} \end{pmatrix} &\leftarrow \begin{pmatrix} \cos \phi_i' & -\sin \phi_i' \\ \sin \phi_i' & \cos \phi_i' \end{pmatrix} \begin{pmatrix} x_{\bar{m}} \\ y_{\bar{m}} \end{pmatrix} && \triangleright \text{3. rotate} \\
\begin{pmatrix} x_{\bar{m}} \\ y_{\bar{m}} \end{pmatrix} &\leftarrow \begin{pmatrix} \alpha_x x_{\bar{m}} \\ \alpha_y y_{\bar{m}} \end{pmatrix} && \triangleright \text{4.a. unstretch} \\
\begin{pmatrix} x_{\bar{m}} \\ y_{\bar{m}} \end{pmatrix} &\leftarrow \begin{pmatrix} x_{\bar{m}} + \bar{x}_{\bar{m}} \\ y_{\bar{m}} + \bar{y}_{\bar{m}} \end{pmatrix} && \triangleright \text{4.b. unshear} \\
\omega_m &\leftarrow \frac{(-x_{\bar{m}}, -y_{\bar{m}}, 1)}{\sqrt{x_{\bar{m}}^2 + y_{\bar{m}}^2 + 1}} && \triangleright \text{5. compute normal}
\end{aligned}$$

11.6 Résultats et discussion

11.6.1 Validation

Pour valider une stratégie d'importance sampling, il faut vérifier que l'estimateur converge bien vers l'intégrale à évaluer. Même si les équations et les dérivations sont correctes, nous ne sommes jamais à l'abri d'une erreur de dérivation ou d'un bug d'implémentation. Pour valider expérimentalement notre modèle et notre implémentation, nous avons utilisé le test du χ^2 disponible dans Mitsuba. Le code que nous distribuons (voir l'annexe F) pour les distributions de Beckmann et GGX passe ce test, ce qui prouve que notre estimateur est consistant.

11.6.2 Analyse de convergence

Les figures 11.11 et 11.12 comparent la convergence de l'importance sampling avec la méthode précédente à la convergence obtenue avec notre méthode dans la configuration pour, respectivement, le même temps de calcul ou le même nombre d'échantillons. Dans la figure 11.11, nous observons qu'utiliser du multiple importance sampling permet d'accélérer la convergence des deux techniques, mais notre technique reste la plus efficace. Dans les deux dernières lignes, le matériau a un profil de micro-surface en V-cavités. La technique précédente ne souffre pas de poids d'échantillons très élevés (ils sont bornés à 2) mais notre technique converge quand même plus rapidement. C'est parce que nous ne générons pas de normales auto-masquées de poids nul.

Plus de comparaisons et de résultats sont disponibles dans l'annexe F.

11.6.3 Les cas très difficiles avec la méthode précédente

La figure 11.13 montre une configuration où les *fireflies artefacts* sont toujours présents avec la technique précédente même avec 4000 échantillons par pixel.

Les cas les plus difficiles pour la technique précédente sont ceux où la scène tend à produire beaucoup de *multiple scattering* (rebonds multiples) sur le matériau à micro-facettes. Cela est intuitif car nous avons vu que les artefacts les plus persistants sont dus aux échantillons de poids arbitrairement élevés. Or, plus un *light path* (chemin de lumière) interagit avec le matériau à micro-facettes, plus les chances de générer de tels échantillons augmentent. Pire, si le même chemin génère deux échantillons aux poids élevés, leurs poids sont multipliés, ce qui augmente encore plus la variance. Pour démontrer cet effet, nous comparons les configurations proposées dans les figures 11.14 et 11.15 : les deux configurations ont les mêmes conditions d'éclairage et les mêmes matériaux, seul le cube présent dans la première figure a été remplacé par un cube fractal dans la seconde figure. Dans la première figure, nous observons que notre méthode produit un résultat sensiblement meilleur avec moins de variance aux angles rasants, comme nous pouvions nous y attendre. Mais la différence de convergence n'est pas non plus d'un ordre de grandeur de différence. C'est parce que cette scène ne contient presque que du *single scattering* : la lumière interagit presque partout seulement une fois avec le cube conducteur. La probabilité de générer des échantillons qui introduisent une très forte variance est donc relativement faible. Dans la seconde figure, nous avons remplacé le cube par une forme fractale qui présente l'intérêt de posséder de nombreuses cavités. Lorsque la lumière y entre, elle rebondit de nombreuses fois avant de pouvoir en ressortir. C'est donc un très bon test pour observer du multiple scattering. Nous pouvons voir dans cette scène que même avec plus de 8000 échantillons par pixel la technique précédente produit une image très fortement bruitée, en particulier dans les cavités de la fractale où la probabilité d'obtenir des fireflies artefacts est très importante à cause du multiple scattering.

Ceci montre que dans les scènes à fort multiple scattering, il est extrêmement important d'avoir une technique d'importance sampling robuste. Notre technique qui produit des échantillons à poids bornés à 1, ne souffre pas de ce problème. Dans cette scène difficile, avec 8 fois moins d'échantillons par pixel et 6 fois moins de temps de calcul nous obtenons une image qui contient beaucoup moins de variance et aucun artefact.

11.6.4 Les performances

Il est à noter que, pour le modèle de Smith, l'implémentation de notre algorithme d'échantillonnage demande plus d'instructions que celui de la technique précédente et consomme plus de temps de calcul. C'est la raison pour laquelle nous avons proposé des résultats à temps de calcul égaux. En moyenne, il est environ 20% moins rapide, mais produit quand même des résultats de meilleure qualité.

Toutefois, la plupart des scènes de test que nous avons utilisées pour tester notre méthode sont relativement simples : les scènes ne contiennent pas beaucoup de triangles ni de textures. De fait, le *bottleneck* (goulot d'étranglement) se trouve probablement dans l'évaluation des BRDFs et dans leur échantillonnage. Avoir une technique plus coûteuse à cet endroit-là du pipeline affecte significativement les performances. Nous pouvons imaginer que, dans des scènes plus complexes avec énormément de géométrie et de texture, le bottleneck du rendu se trouverait ailleurs, par exemple dans le chargement des textures, dans les tests d'intersection avec la géométrie, etc. et que l'impact d'utiliser notre méthode serait moins important pour les performances. Nos tests de comparaison nous placent donc dans des cas défavorables à notre méthode où elle s'avère quand même supérieure à la technique précédente. Nous ne pouvons donc qu'obtenir des comparaisons au moins aussi bonnes, voire encore meilleures, sur des scènes plus difficiles.

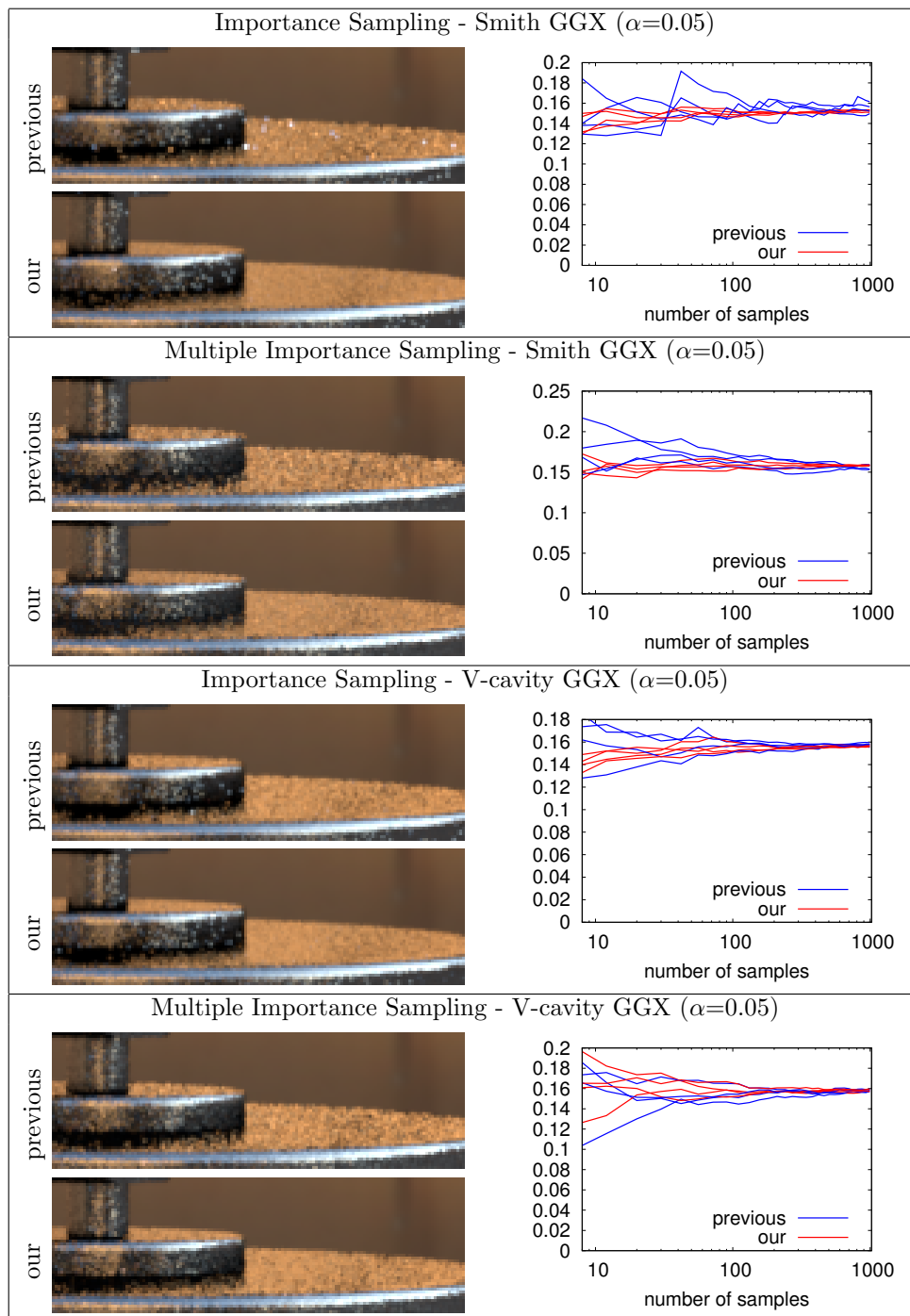


FIGURE 11.11 – Comparaison de la convergence entre la technique précédente et la nôtre. Les images sont calculées avec 8 échantillons par pixel. Les graphes montrent la convergence de plusieurs instances de calcul pour le même pixel pour les deux techniques. La configuration est celle présentée dans la figure 11.2.

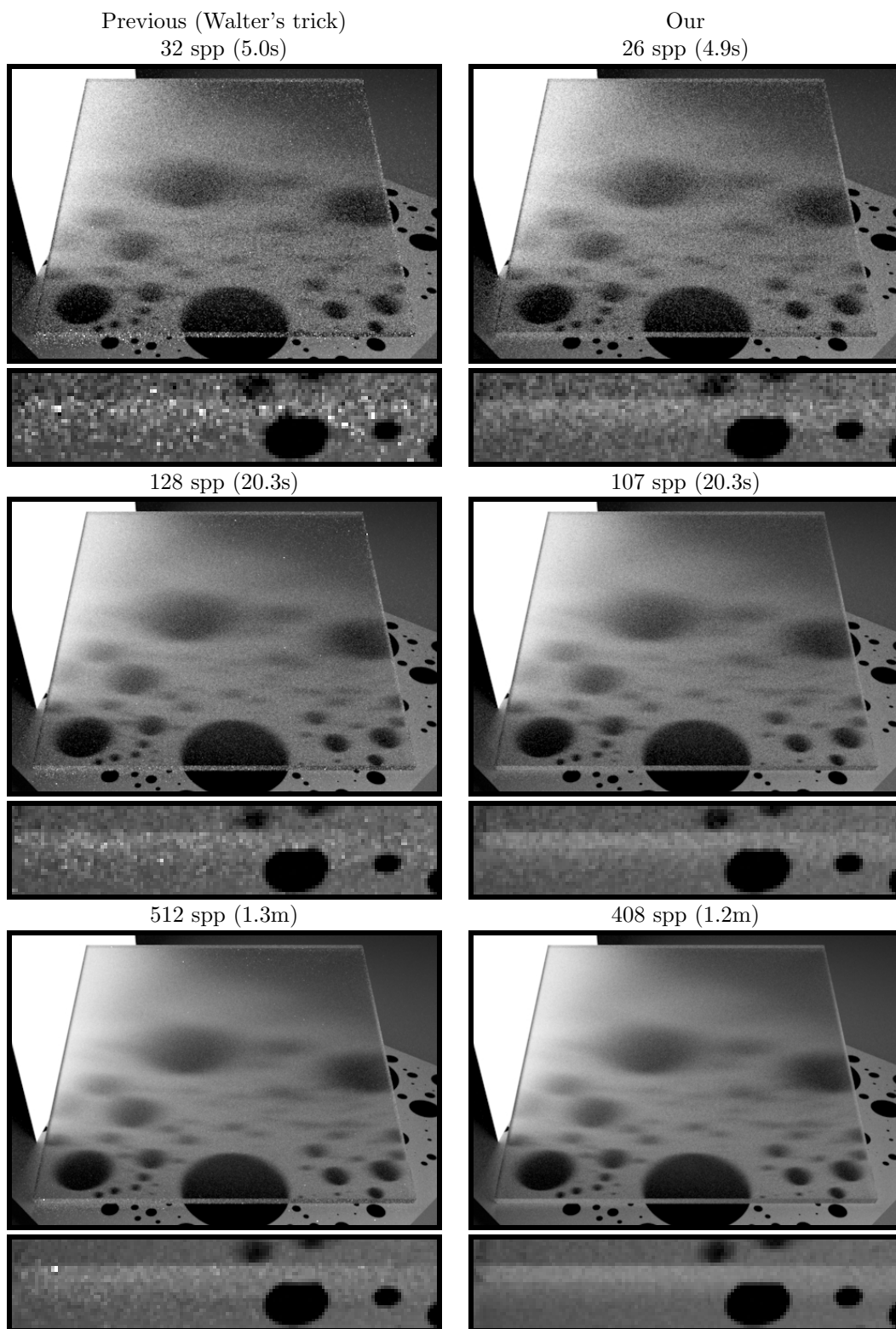


FIGURE 11.12 – Un plateau fait d'un matériau diélectrique ($n = 1.5$) avec une rugosité anisotrope (Beckmann $\alpha_x = 0.05$, $\alpha_y = 0.40$) dont le profil de micro-surface est celui de Smith. Notre méthode converge plus vite pour le même budget en temps de calcul.

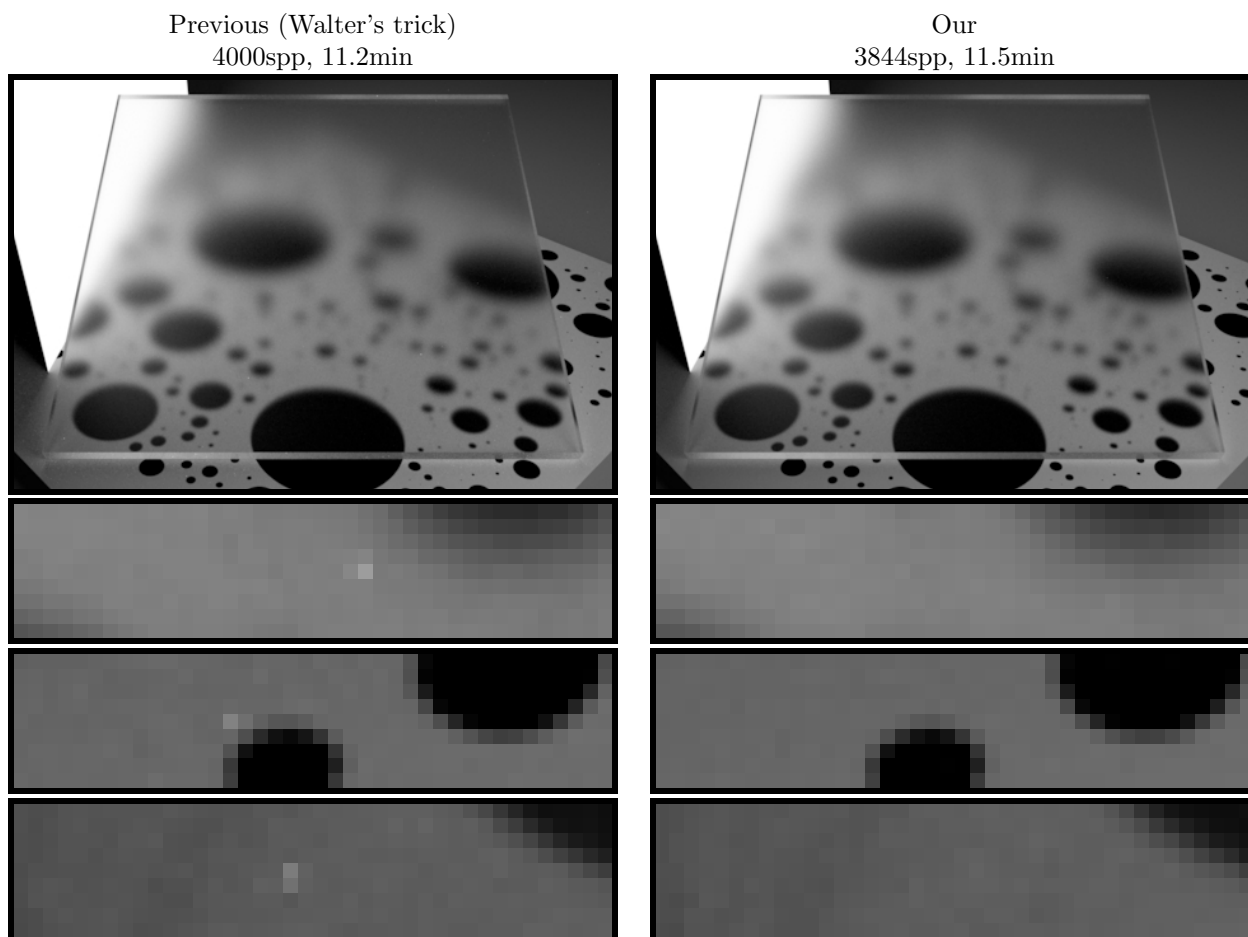


FIGURE 11.13 – Un plateau fait d'un matériau diélectrique ($n = 1.5$) avec une rugosité isotrope (Beckmann $\alpha = 0.10$) dont le profil de micro-surface est celui de Smith calculé avec 4000 échantillons par pixel. Les fireflies artefacts sont persistants avec la technique précédente même avec un grand nombre d'échantillons.

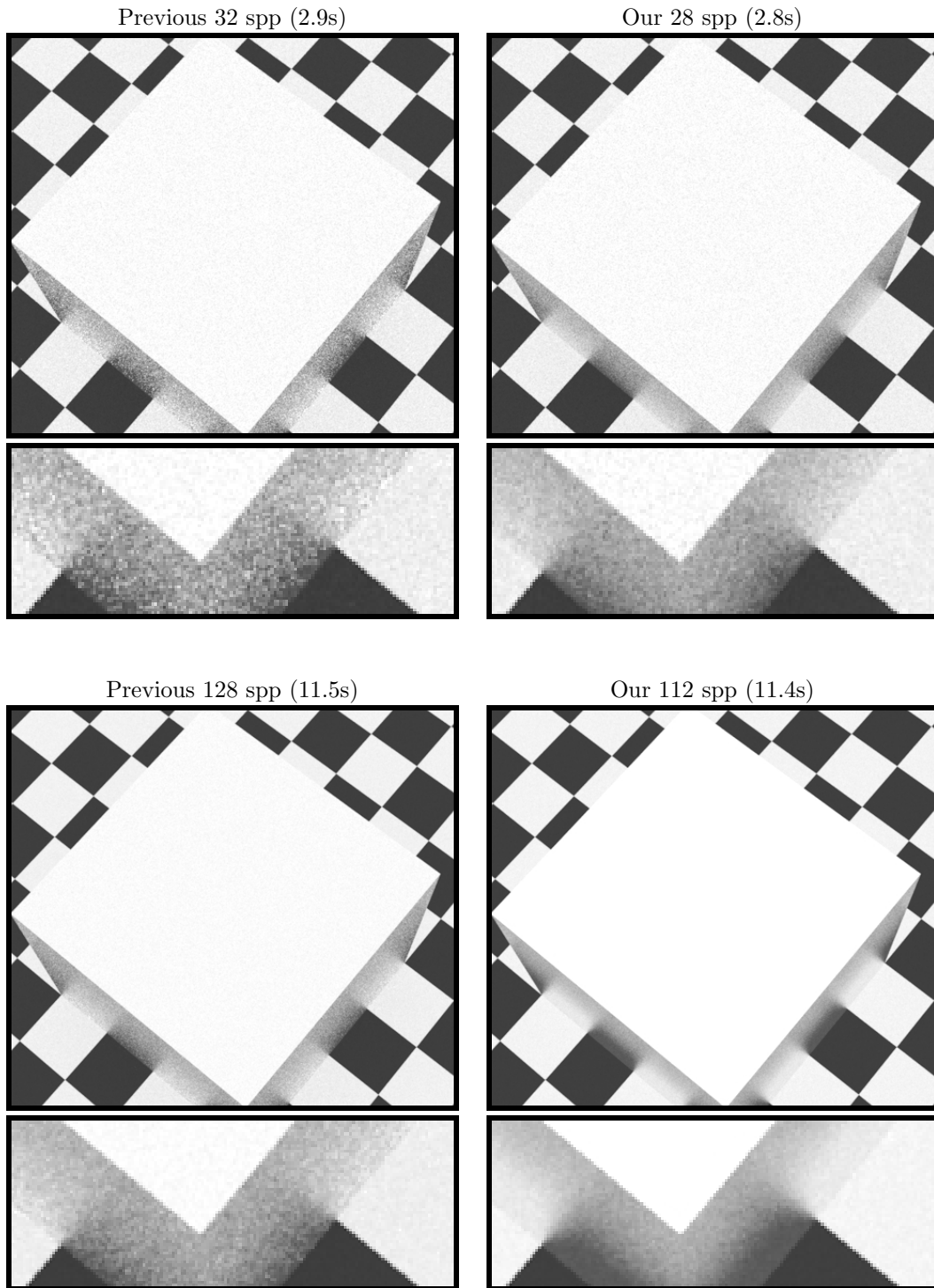


FIGURE 11.14 – Un cube fait d'un matériau conducteur ($n = 1.5$) avec une rugosité isotrope (GGX $\alpha = 0.15$) dont le profil de micro-surface est celui de Smith. Notre méthode converge plus vite pour le même budget en temps de calcul.

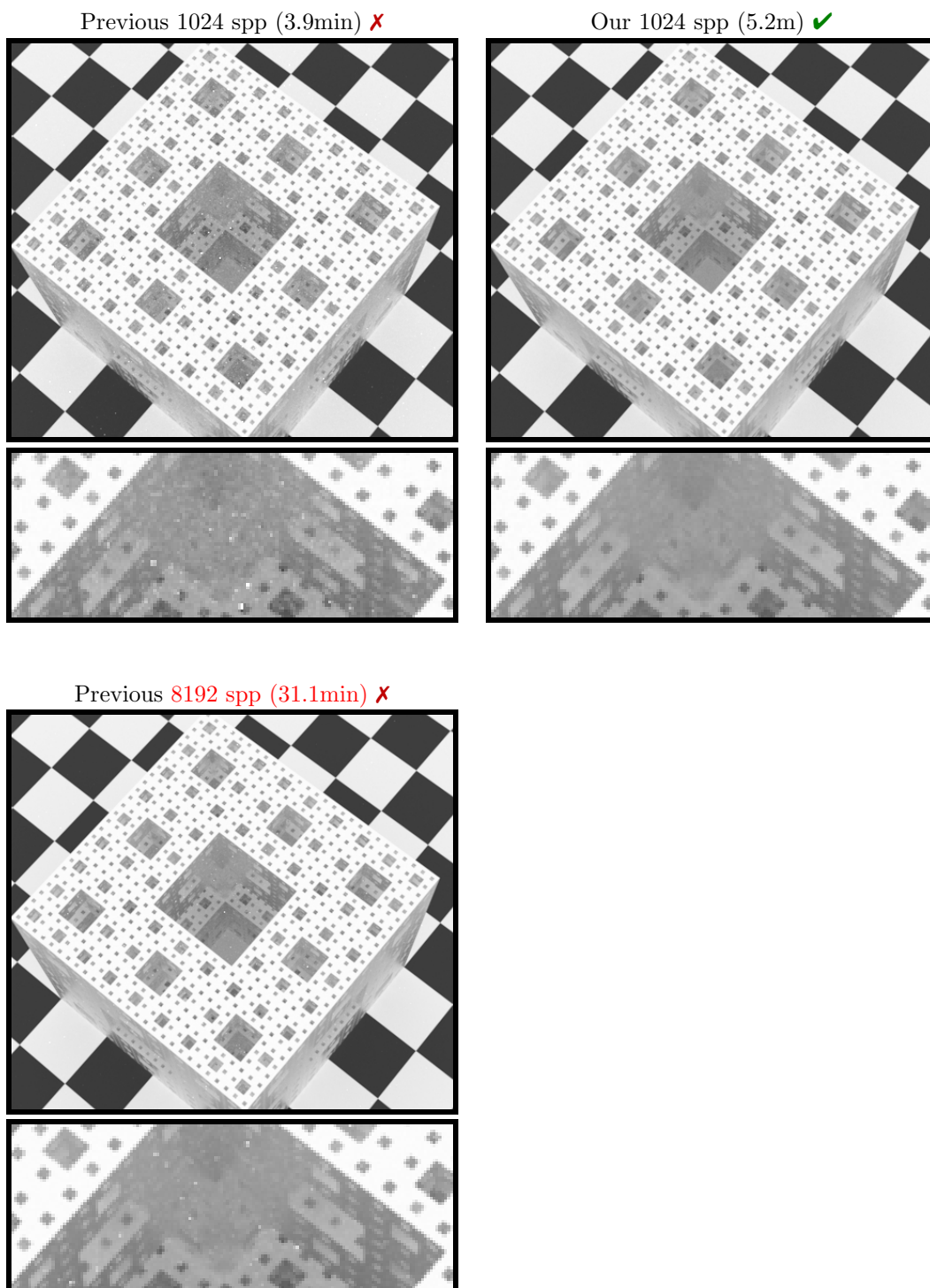


FIGURE 11.15 – Un cube de Sierpinski fait d'un matériau conducteur ($n = 1.5$) avec une rugosité isotrope ($GGX \alpha = 0.15$) dont le profil de micro-surface est celui de Smith. Cette scène produit beaucoup de multiple scattering à l'intérieur de la forme fractale. Même après plus de 8000 échantillons par pixel la technique précédente contient toujours des artefacts très visibles. Notre méthode produit une image utilisable après 1000 échantillons par pixel, soit 8 fois moins, et 6 fois moins de temps de rendu. (Zoomer dans le PDF pour voir les artefacts dans les cavités de la fractale avec la technique précédente.)

11.7 Résumé du chapitre

Points clés du chapitre

- Le modèle de BSDFs à micro-facettes est construit sur la distribution des normales visibles.
- La technique d'importance sampling précédente, qui est la plus répandue, utilise la distribution de normales pour générer des échantillons, sans tenir compte de leur visibilité.
- Ne pas tenir compte de la visibilité peut générer des échantillons de poids arbitrairement élevé, responsables des *fireflies artefacts* présents dans les images, et gâcher jusqu'à la moitié de l'espace d'échantillonnage en générant des normales auto-masquées dont la contribution est nulle.
- Nous proposons d'utiliser la distribution de normales *visibles* pour générer les échantillons. De fait, notre technique d'importance sampling reproduit exactement ce qui est modélisé dans la théorie des micro-facettes à l'exception de la probabilité d'ombrage des directions sortantes.
- Le poids des échantillons générés avec notre technique représente la probabilité d'ombrage dont nous ne tenons pas compte pour la génération des échantillons. Le poids est toujours dans $[0, 1]$ (puisque l'ombrage est une probabilité), ce qui évite les artefacts dus aux poids trop élevés. De plus, par construction, nous ne générons jamais de normales auto-masquées et ne gaspillons pas l'espace d'échantillonnage. Grâce à ces propriétés, notre technique fait converger le résultat plus vite que la technique précédente.
- Nous instancions cette technique pour le modèle à V-cavités et montrons que cela peut être réalisé simplement en échangeant les normales échantillonnées avec des probabilités proportionnelles à leurs visibilités.
- Nous instancions cette technique pour le modèle de Smith pour lequel les dérivations sont plus compliquées en général. Nous surmontons ce problème en exploitant l'invariance par étirement de la distribution des pentes visibles. Grâce à cette propriété d'invariance, si une stratégie d'échantillonnage est connue pour une seule micro-surface, elle peut être utilisée pour toutes les micro-surfaces de la même classe avec n'importe quelle rugosité ou anisotropie. Nous proposons une technique d'échantillonnage analytique pour les distributions de Beckmann et GGX.

A voir

- L'annexe F propose la dérivation de la stratégie de sampling analytique pour les distributions de Beckmann et GGX avec le code C++ associé, ainsi que de nombreux résultats de comparaisons supplémentaires en graphiques et en images.
- Le papier EGSR [\[HD14\]](#), ses deux supplemental materials et les slides de la présentation.

Troisième partie

Conclusion

Chapitre 12

Conclusion

12.1 Résumé des contributions

Dans cette thèse, nous avons étudié le filtrage de l'apparence pour le rendu multi-échelles efficace des surfaces détaillées. Nos contributions se divisent en deux parties : un framework théorique pour modéliser les surfaces multi-échelles et la mise au point de représentation de modèle de calcul pour les évaluer en pratique.

Notre framework théorique est essentiellement basé sur la théorie des micro-facettes. Nous avons commencé par revisiter cette théorie afin de mieux en comprendre les fondements et pouvoir l'étendre en partant sur des bases solides. En partant de propriétés géométriques (la conservation de l'aire projetée visible), nous avons redérivé les différents termes qui interviennent dans l'équation d'illumination locale, nous avons mis au point des critères pour les valider, et nous avons étudié les différentes manières dont ils peuvent être corrélés entre eux.

Pour représenter et rendre des surfaces multi-échelles en pratique, il faut instancier les équations génériques du modèle. Nous avons motivé et discuté des choix techniques, en particulier le choix d'une représentation. Nous avons choisi de nous baser sur une représentation des surfaces multi-échelles de type height map avec des statistiques gaussiennes. Nous avons vu que cette représentation a de nombreuses propriétés la rendant particulièrement facile à construire et à stocker dans des textures et que, par ailleurs, elle se traite particulièrement bien. Nous avons également montré comment rendre cette représentation compatible avec l'animation des modèles.

Enfin, nous avons étudié l'intégration du modèle de matériau avec l'éclairage dans deux contextes opposés, l'un étant un dédié aux applications temps-réel, l'autre au rendu dans un moteur de type Monte Carlo avec de l'importance sampling.

12.2 Limites et travaux futurs à court terme

Nous avons vu dans le chapitre d'introduction que certains modèles très poussés existent pour des cas particuliers spécialisés (l'océan, la forêt, etc.). Nous considérons les travaux présentés dans ce manuscrit comme un premier pas vers un modèle plus généraliste, plus ouvert. Mais le problème du filtrage de l'apparence et du rendu multi-échelles est loin d'être résolu. Voici les limites immédiates auxquelles se heurte notre modèle en l'état actuel et des idées pour les repousser.

Les height maps. Nous avons mis au point un modèle basé sur la height map comme primitive de surface multi-échelles. Cette primitive ne permet donc pas de modéliser des surfaces qu'en 2.5D contrairement aux *displacement maps*, dont les déplacements sont des vecteurs 3D et non des valeurs scalaires de hauteurs. Mais dans de nombreux cas, cette limitation est facilement contournable. En effet, si de nombreux logiciels d'édition offrent des outils pour modéliser des displacement maps, les surfaces modélisées par les artistes sont souvent isomorphes à des height maps, dans le sens où les surfaces détaillées résultantes peuvent se modéliser

par des height maps définies par rapport à un plan moyen. Dans ce cas, la conversion displacement map \rightarrow height map peut se faire en calculant une simple projection orthogonale de la surface déplacée dans son plan moyen. Par contre, toutes les displacement maps ne sont pas isomorphes à des height maps. C’est le cas lorsqu’aucun opérateur de projection ne produit une transformation bijective, ou, plus intuitivement, lorsqu’il existe des normales qui sont “à l’envers” (avec $z < 0$) et qui ne peuvent pas être représentées par une pente par rapport à un plan moyen. Dans ce cas, notre modèle de filtrage basé sur l’espace des pentes n’est pas applicable.

Les statistiques gaussiennes. Nous avons choisi de représenter les surfaces multi-échelles et ses matériaux avec des distributions de pentes à statistiques gaussiennes. Ces statistiques sont applicables sur n’importe quel type de height map mais ne représentent pas toujours avec précision l’apparence multi-échelles de celle-ci. En effet, les données ne sont pas toujours à statistiques gaussiennes. Il pourrait être intéressant d’explorer d’autres types de distributions, plus riches, mais tout en gardant les bonnes propriétés des gaussiennes : linéairement filtrable, coût mémoire raisonnable, etc. L’approximation gaussienne suppose aussi une entrée rapide dans la loi des grands nombres. Une étude récente [YHJ⁺14] montre que les distributions de normales contiennent certains patterns importants pour l’apparence de certains matériaux, en particulier certains effets de brillance hautes-fréquences disparaissent lorsque les détails les plus fins de la distribution de normales ne sont pas préservés. Mais ces effets n’existent que pour des matériaux très spéculaires sous des éclairages très hautes-fréquences. Il serait intéressant d’étudier plus en profondeur comment choisir le bon niveau de filtrage en fonction des conditions d’éclairage et des matériaux, c’est-à-dire de trouver à partir de quel moment la loi des grands nombres s’applique de manière à pouvoir garder la propriété de scalabilité mais en conservant l’apparence des détails les plus fins.

Les modèles de matériaux. Les exigences pour les modèles de matériaux évoluent très vite et ont déjà dépassé le modèle basé sur des statistiques gaussiennes (distribution de Beckmann) que nous proposons d’utiliser. En effet, il est à noter par exemple que, ces deux dernières années, presque toute la communauté du rendu temps-réel (en particulier les industriels du jeu vidéo) a mis de côté les statistiques gaussiennes pour passer sur des modèles à micro-facettes basés sur la distribution GGX, plus proche de données mesurées et plus réaliste selon les artistes. Pour rendre nos idées compatibles avec cette demande, une manière simple de procéder serait d’utiliser la même représentation (les moments d’une distribution de pentes multi-échelles) mais de les utiliser comme paramètres pour une distribution GGX plutôt que pour une distribution de Beckmann. Etant donné que la plupart de nos résultats sont dérivés de manière indépendante de la distribution (modèle décentré, importance sampling, etc.), changer simplement la forme de la distribution a un impact assez faible sur notre modèle en général.

En plus du choix de la distribution de micro-facettes, nous n’avons pas non plus exploré tous les types de micro-matériaux. A titre d’exemple, nous n’avons pas parlé de la représentation des matériaux multi-couches, utilisés presque partout dans les films et les jeux vidéos photo-réalistes. En général, ces matériaux multi-couches sont mathématiquement de simples combinaisons linéaires de matériaux mono-couche à micro-facettes spéculaires ou diffuses. Ce serait donc une extension assez simple à incorporer à notre modèle.

Par ailleurs, nous avons aussi vu que les modèles de BRDFs à micro-facettes ne modélisent que le premier rebond sur l’interface spéculaire, les rebonds suivants étant supprimés par la fonction d’ombrage. La technique d’importance sampling que nous proposons pourrait être utilisée pour importance sampler les rebonds suivants. En effet, puisque nous connaissons la probabilité d’observer des rebonds multiples (donnée par la fonction d’ombrage), et que nous connaissons la direction des rayons qui intersectent potentiellement plusieurs fois la micro-surface, nous pouvons simplement importance sampler les micro-facettes associées à chaque rebond et appliquer un opérateur de transport (comme *reflect*) à chaque fois jusqu’à ce que les rayons finissent par quitter la surface. De cette manière, nous pouvons importance sampler des matériaux avec plusieurs rebonds et nous pourrions les évaluer dans un moteur qui fait du *forward pathtracing*. En revanche, comme nous ne savons pas évaluer ces matériaux (nous n’avons pas de forme close pour la BRDF), nous ne pourrions pas les rendre dans un moteur qui fait des calculs en *backward*. Il serait intéressant de dériver une forme close pour ces matériaux complexes ou de trouver une astuce pour les évaluer efficacement.

Dans un travail récent, Wenzel et al. [JDJM14] proposent des outils mathématiques pour modéliser et évaluer ce genre de matériaux.

L’hypothèse de planarité. Nous avons proposé une manière de repousser légèrement la limite imposée par l’hypothèse de planarité en utilisant une astuce pour représenter et intégrer la courbure à notre modèle de filtrage. La manière dont nous l’intégrons est assez grossière et il serait intéressant d’étudier plus en détail comment faire le lien proprement entre la courbure et la distribution de normales. En particulier, comment mieux évaluer les paramètres de la distribution de normales dus à la courbure avec, notamment, leur anisotropie. Ceux-ci doivent pouvoir se déduire du *shape operator* associé à la courbure qui est évaluable depuis les dérivées des normales [GWBO12].

12.3 Perspectives à long terme

Pour en revenir à l’objectif général du rendu multi-échelles, c’est-à-dire la mise au point du “Modèle Ultime” de filtrage qui serait capable de modéliser toutes les composantes d’une scène en même temps quelles que soient leurs natures, notre limitation principale évidente est que notre framework est adapté au rendu des surfaces uniquement. Et cela vient avec certaines limitations assez fortes, auxquelles nous nous heurtons très vite.

Représentation multi-échelles des volumes. En effet, nous avons vu par exemple que la courbure géométrique constitue déjà un problème important en pratique avec des modèles très détaillés. Mais ce n’est pas le seul problème lié aux surfaces. Par exemple, que faire aux silhouettes où l’empreinte anisotrope du pixel est tellement allongée que l’hypothèse de localité n’est plus valable? Dans les cas extrêmes de silhouettes, le contenu de ce qui se projette dans le même pixel ressemble plus à une soupe d’éléments volumiques qu’à une surface continue et bien paramétrée.

Par ailleurs, nous arrivons aussi très vite dans les cas où les objets ne sont eux-mêmes plus représentables par des surfaces et des textures mais deviennent de nature intrinsèquement volumique. C’est le cas par exemple pour un arbre vu de loin : ses feuilles se projettent par dizaines au sein du même pixel et ne sont plus représentables par une surface avec une BRDF. Il en va de même pour d’autres représentations intrinsèquement non surfaciques : la fourrure, l’herbe, les grillages, les gaz, etc. Un modèle ne pouvant représenter que des surfaces se retrouve donc très vite coincé par la nature même des surfaces.

L’étape suivante naturelle est donc la mise au point d’un framework similaire dédié à représenter les volumes multi-échelles. Il pourrait se baser par exemple sur l’équivalent volumique des micro-facettes : les micro-flakes [JAM⁺10]. J’ai eu l’occasion de commencer à travailler sur un tel projet lors de mes passages chez NVIDIA et Weta Digital. Mais il reste beaucoup de choses à faire, autant sur le design théorique du modèle que sur le choix d’une représentation utilisable en pratique avec les propriétés que nous avons suggérées, sans compter les exigences des utilisateurs en terme de choix de matériaux, d’API d’utilisation, etc. Par ailleurs, les volumes présentent des difficultés supplémentaires comparés aux surfaces. En effet, la gestion de la visibilité est différente : les corrélations d’occlusion et d’opacité sont très difficiles à représenter et à filtrer le long d’un rayon, mais sont très importantes pour préserver l’apparence. Au début de ma thèse, j’avais commencé à travailler sur ce problème de filtrage de la visibilité [HN12] avant de me concentrer exclusivement sur le filtrage du shading, qui à lui tout seul constituait déjà un problème difficile. Mais il serait intéressant d’y revenir.

L’unification des modèles multi-échelles. Et quand bien même il existerait des frameworks aboutis dédiés aux surfaces et d’autres aux volumes, le pas suivant serait de chercher à les unifier, problème qui est loin d’être facile : à partir de quand une empreinte surfacique peut-elle se représenter par un volume? Comment la fonction de phase émerge-t-elle de la BRDF? Et réciproquement? Et comment gérer la transition?

La formulation dans le framework moderne d’illumination globale. Ces dernières années ont vu un regain d’intérêt pour le *core rendering* avec l’unification de différentes techniques autour de la formulation

du rendu dans l'espace des chemins [Vea98], en particulier, avec des techniques comme *Vertex Connection and Merging* [GKDS12, HPJ12] et ses extensions récentes [KGH⁺14]. Ces méthodes sont basées sur la notion de *footprint* : le moteur de rendu ne calcule plus des rayons ponctuels, mais des rayons qui ont une certaine épaisseur. Le fait d'avoir un footprint permet d'effectuer des recherches beaucoup plus efficacement dans l'espace des chemins, mais celui-ci introduit un biais dans le résultat. Il y a donc un compromis biais/performances à ajuster arbitrairement. Mais avec la notion de footprint vient naturellement la notion de filtrage et de niveaux de détails. Il serait donc intéressant d'introduire formellement le rendu multi-échelles dans cette formulation, puisque le choix du niveau de détails est en quelque sorte le problème dual du choix du footprint.

12.4 Le Modèle Ultime de filtrage

Si cette thèse n'a évidemment pas résolu le problème du rendu en général, elle a apporté des éléments qui donnent à penser que le rendu multi-échelles garantissant des images de qualité est possible, et cela en passant par une étude rigoureuse des modèles et des choix de représentation et de calcul. Nous espérons donc qu'appliquer la même méthodologie permettra de repousser les limites de ce framework et d'en construire d'autres, pour nous approcher petit à petit du "Modèle Ultime" de filtrage. Ce Graal d'unification, que nous avons évoqué à plusieurs reprises, est probablement utopique. Mais ce n'est pas grave, car, même sans l'atteindre, le simple fait de nous en approcher nous permet d'introduire plus d'intelligence et d'efficacité dans les techniques d'aujourd'hui et d'imaginer celles de demain.

Bibliographie

- [Ama92] John Amanatides. Algorithms for the detection and elimination of specular aliasing. In *Proceedings of the Conference on Graphics Interface '92*, pages 86–93, 1992.
- [APS00] Michael Ashikmin, Simon Premože, and Peter Shirley. A microfacet-based BRDF generator. In *Proceedings of ACM SIGGRAPH' 00*, pages 65–74, 2000.
- [AS00] Michael Ashikhmin and Peter Shirley. An anisotropic phong BRDF model. *Journal of Graphics Tools*, 5 :25–32, 2000.
- [Ash07] Michael Ashikhmin. Distribution-based BRDFs. Technical report, 2007.
- [Bak11] Dan Baker. Spectacular specular – LEAN and CLEAN specular highlights. In *Proc. Game Developer Conference 2011*, 2011.
- [BB03] Bourlier and G. Berginc. Shadowing function with single reflection from anisotropic gaussian rough surface. application to gaussian, lorentzian and sea correlations. *Waves in Random and Complex Media*, 13(1) :27–58, mar 2003.
- [BBS02] C. Bourlier, G. Berginc, and Joseph Saillard. One- and two-dimensional shadowing functions for any height and slope stationary uncorrelated surface in the monostatic and bistatic configurations. *IEEE Trans. on Antennas and Propagation*, 50(3) :312–324, Mar 2002.
- [BL08] Brent Burley and Dylan Lacewell. Ptex : Per-face texture mapping for production rendering. In *Proceedings of the Eurographics Symposium on Rendering*, pages 1155–1164, 2008.
- [Bli77] James F. Blinn. Models of light reflection for computer synthesized pictures. *ACM SIGGRAPH Comput. Graph.*, 11(2) :192–198, July 1977.
- [BM93] Barry G. Becker and Nelson L. Max. Smooth transitions between bump rendering algorithms. In *Proceedings of ACM SIGGRAPH '93*, pages 183–190, 1993.
- [BMWM06] S. Bergner, T. Moller, D. Weiskopf, and D.J. Muraki. A spectral analysis of function composition and its implications for sampling in direct volume visualization. *IEEE Trans. on Visualization and Computer Graphics*, 12(5) :1353–1360, 2006.
- [BN12a] Eric Bruneton and Fabrice Neyret. A Survey of Non-linear Pre-filtering Methods for Efficient and Accurate Surface Shading. *IEEE Transactions on Visualization and Computer Graphics*, 18(2) :242–260, February 2012.
- [BN12b] Eric Bruneton and Fabrice Neyret. Real-time Realistic Rendering and Lighting of Forests. *Computer Graphics Forum*, 31(2pt1) :373–382, May 2012.
- [BNH10] Eric Bruneton, Fabrice Neyret, and Nicolas Holzschuch. Real-time Realistic Ocean Lighting using Seamless Transitions from Geometry to BRDF. *Computer Graphics Forum*, 29(2) :487–496, 2010.
- [Bro80] G.S. Brown. Shadowing by non-gaussian random surfaces. *IEEE Trans. on Antennas and Propagation*, 28(6) :788–790, Nov 1980.
- [BS63] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. International series of monographs on electromagnetic waves. Pergamon Press, 1963.

- [BSB00] C. Bourlier, J. Saillard, and G. Berginc. Effect of correlation between shadowing and shadowed points on the wagner and smith monostatic one-dimensional shadowing functions. *IEEE Trans. on Antennas and Propagation*, 48(3), mar 2000.
- [BSH12] M. Bagher, Mahdi, Cyril Soler, and Nicolas Holzschuch. Accurate fitting of measured reflectances using a Shifted Gamma micro-facet distribution. *Computer Graphics Forum*, 31(4) :1509–1518, June 2012.
- [Bun05] Michael Bunnell. Adaptive tessellation of subdivision surfaces with displacement mapping. In *GPU Gems 2*. 2005.
- [CK07] Mark Colbert and Jaroslav Krivánek. GPU-based importance sampling. In *GPU Gems 3*, chapter 20. Addison-Wesley, 2007.
- [CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1) :7–24, January 1982.
- [Dac06] Carsten Dachsbacher. *Interactive Terrain Rendering : Towards Realism with Procedural Models and Graphics Hardware*. PhD thesis, 2006.
- [Deb05] Paul Debevec. Image-based lighting. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, 2005.
- [DHI⁺13] Jonathan Dupuy, Eric Heitz, Jean-Claude Iehl, Pierre Poulin, Fabrice Neyret, and Victor Ostromoukhov. Linear Efficient Antialiased Displacement and Reflectance Mapping. 32(6), 2013.
- [Fou92] Alain Fournier. Normal distribution functions and multiple surfaces. In *Graphics Interface '92 Workshop on Local Illumination*, pages 45–52, Vancouver, BC, Canada, 11 May 1992.
- [FW94] Frederick Fisher and Andrew Woo. Graphics gems iv. chapter R.E Versus N.H Specular Highlights, pages 388–400. Academic Press Professional, Inc., San Diego, CA, USA, 1994.
- [GKDS12] Iliyan Georgiev, Jaroslav Krivánek, Tomáš Davidovič, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.*, 31(6) :192 :1–192 :10, November 2012.
- [Gre07] Chris Green. Efficient self-shadowed radiosity normal mapping. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, pages 1–8, 2007.
- [GWBO12] Wesley Griffin, Yu Wang, David Berrios, and Marc Olano. Real-time gpu surface curvature estimation on deforming meshes and volumetric data sets. *IEEE Trans. on Visualization and Computer Graphics*, 18(10) :1603–1613, 2012.
- [HBP13] Eric Heitz, Christophe Bourlier, and Nicolas Pinel. Correlation effect between transmitter and receiver azimuthal directions on the illumination function from a random rough surface. *Waves in Random and Complex Media*, 23(3) :318–335, 2013.
- [HCK⁺99] John C. Hart, Nate Carr, Masaki Kameya, Stephen A. Tibbitts, and Terrance J. Coleman. Antialiased parameterized solid texturing simplified for consumer-level hardware implementation. In *Proceedings of ACM SIGGRAPH / Eurographics Workshop on Graphics Hardware*, pages 45–53, 1999.
- [HD14] Eric Heitz and Eugene D'Eon. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, 2014.
- [Hei14] E. Heitz. Understanding the masking-shadowing function in microfacet-based BRDFs. *Journal of Computer Graphics Techniques*, 3(2), 2014. To appear.
- [HN12] Eric Heitz and Fabrice Neyret. Representing appearance and pre-filtering subpixel data in sparse voxel octrees. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics*, pages 125–134, 2012.
- [HNPN13a] Eric Heitz, Derek Nowrouzezahrai, Pierre Poulin, and Fabrice Neyret. Filtering Color Mapped Textures and Surfaces. In *Proceedings of ACM Symposium on Interactive 3D Graphics*, pages 129–136, 2013.

- [HNPN13b] Eric Heitz, Derek Nowrouzezahrai, Pierre Poulin, and Fabrice Neyret. Filtering Non-Linear Transfer Functions on Surfaces. *IEEE Trans. on Visualization and Computer Graphics*, November 2013.
- [HPJ12] Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. A path space extension for robust light transport simulation. *ACM Trans. Graph.*, 31(6) :191 :1–191 :10, November 2012.
- [HSB⁺12] Markus Hadwiger, Ronell Sicut, Johanna Beyer, Jens Krüger, and Torsten Möller. Sparse pdf maps for non-linear multi-resolution image operations. *ACM Trans. Graph.*, 31(6) :133 :1–133 :12, 2012.
- [HSRG07] Charles Han, Bo Sun, Ravi Ramamoorthi, and Eitan Grinspun. Frequency domain normal map filtering. In *Proceedings of ACM SIGGRAPH '07*, 2007.
- [HSS98] Wolfgang Heidrich, Philipp Slusallek, and Hans-Peter Seidel. Sampling procedural shaders using affine arithmetic. *ACM Trans. Graph.*, 17(3) :158–176, 1998.
- [HTSG91] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. In *Proceedings of ACM SIGGRAPH '91*, pages 175–186, 1991.
- [HW10] Ralf Habel and Michael Wimmer. Efficient irradiance normal mapping. In *Proceedings of ACM Symposium on Interactive 3D Graphics*, pages 189–195, 2010.
- [Jäc05] P. Jäckel. A note on multivariate Gauss-Hermite quadrature, 2005. <http://www.jaeckel.org/>.
- [JAM⁺10] Wenzel Jakob, Adam Arbree, Jonathan T. Moon, Kavita Bala, and Steve Marschner. A radiative transfer framework for rendering materials with anisotropic structure. *ACM Trans. Graph.*, 29(4) :53 :1–53 :13, July 2010.
- [JDJM14] Wenzel Jakob, Eugene D’Eon, Otto Jakob, and Steve Marschner. A comprehensive framework for rendering layered materials. *Proceedings of ACM SIGGRAPH '14*, 33(4), 2014.
- [Kaj86] James T. Kajiya. The rendering equation. In *Proceedings of ACM SIGGRAPH '86*, pages 143–150, 1986.
- [KC08] Jaroslav Krivánek and Mark Colbert. Real-time shading with filtered importance sampling. *Proceedings of the Eurographics Symposium on Rendering*, 27(4) :1147–1154, 2008.
- [Kel97] Alexander Keller. Instant radiosity. In *Proceedings of ACM SIGGRAPH '97*, pages 49–56, 1997.
- [KGH⁺14] Jaroslav Krivánek, Iliyan Georgiev, Toshiya Hachisuka, Petr Vévoda, Martin Šik, Derek Nowrouzezahrai, and Wojciech Jarosz. Unifying points, beams, and paths in volumetric light transport simulation. *ACM Trans. Graph.*, 33(4) :1–13, August 2014.
- [KSK01] Csaba Kelemen and Laszlo Szirmay-Kalos. A microfacet based coupled specular-matte BRDF model with importance sampling. In *Eurographics Short Presentations*, 2001.
- [KSKK10] Murat Kurt, László Szirmay-Kalos, and Jaroslav Krivánek. An anisotropic BRDF model for fitting and monte carlo rendering. *ACM SIGGRAPH Comput. Graph.*, 44(1) :3 :1–3 :15, February 2010.
- [Lam60] J.H. Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. 1760.
- [LD11] Ares Lagae and George Drettakis. Filtering solid gabor noise. *ACM Trans. Graph.*, 30(4) :51 :1–51 :6, July 2011.
- [Lew94] Robert R. Lewis. Making shaders more physically plausible. In *In Fourth Eurographics Workshop on Rendering*, pages 47–62, 1994.
- [LFTG97] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *Proceedings of ACM SIGGRAPH '97*, pages 117–126, 1997.
- [LLC⁺10] Ares Lagae, Sylvain Lefebvre, Rob Cook, Tony DeRose, George Drettakis, D.S. Ebert, J.P. Lewis, Ken Perlin, and Matthias Zwicker. A survey of procedural noise functions. *Computer Graphics Forum*, 29(8) :2579–2600, 2010.

- [LLDD09] Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse gabor convolution. In *ACM Trans. Graph.*, pages 54 :1–10, 2009.
- [LWC⁺02] David Luebke, Benjamin Watson, Jonathan D. Cohen, Martin Reddy, and Amitabh Varshney. *Level of Detail for 3D Graphics*. Elsevier Science Inc., 2002.
- [McT04] Gary McTaggart. Half-life 2 / valve source shading. In *Proc. Game Developer Conference 2004*, 2004.
- [MCT⁺05] Wan-Chun Ma, Sung-Hsiang Chao, Yu-Ting Tseng, Yung-Yu Chuang, Chun-Fa Chang, Bing-Yu Chen, and Ming Ouhyoung. Level-of-detail representation of bidirectional texture functions for real-time rendering. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, pages 187–194, 2005.
- [MHH⁺12] Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. Practical physically-based shading in film and game production. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 10 :1–7, 2012.
- [MHM⁺13] S. McAuley, S. Hill, A. Martinez, R. Villemin, M. Pettineo, D. Lazarov, D. Neubelt, B. Karis, C. Hery, Naty Hoffman, and H. Zap Andersson. Physically based shading in theory and practice. In *ACM SIGGRAPH 2013 Courses*, SIGGRAPH '13, pages 22 :1–8, 2013.
- [MPA⁺10] Sylvain Meunier, Romuald Perrot, Lilian Aveneau, Daniel Meneveaux, and Djamchid Ghazanfarpour. Cosine lobes for interactive direct lighting in dynamic scenes. *Computers & Graphics*, 34(6) :767 – 778, 2010. Graphics for Serious Games Computer Graphics in Spain : a Selection of Papers from {CEIG} 2009 Selected Papers from the {SIGGRAPH} Asia Education Program.
- [Mus02] F. K. Musgrave. Fractal solid textures : Some examples. In *Texturing and Modeling : A Procedural Approach*, chapter 15, pages 447–487. Morgan Kaufmann, 2002.
- [NDM05] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental analysis of BRDF models. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, pages 117–126, 2005.
- [Nic65] Fred E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Appl. Opt.*, 4(7) :767–775, Jul 1965.
- [OB10] Marc Olano and Dan Baker. Lean mapping. In *Proceedings of ACM Symposium on Interactive 3D Graphics*, pages 181–188, 2010.
- [OKS03] Marc Olano, Bob Kuehne, and Maryann Simmons. Automatic shader level of detail. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 7–14, 2003.
- [ON94] Michael Oren and Shree K. Nayar. Generalization of Lambert’s reflectance model. In *Proceedings of ACM SIGGRAPH '94*, pages 239–246, 1994.
- [Pea85] Darwyn R. Peachey. Solid texturing of complex surfaces. In *Proceedings of ACM SIGGRAPH '85*, pages 279–286, 1985.
- [Pel05] Fabio Pellacini. User-configurable automatic shader simplification. *ACM Trans. Graph.*, 24(3) :445–452, 2005.
- [Per85] Ken Perlin. An image synthesizer. In *Proceedings of ACM SIGGRAPH '85*, pages 287–296, 1985.
- [PH10] Matt Pharr and Greg Humphreys. *Physically Based Rendering, Second Edition : From Theory To Implementation*. Morgan Kaufmann Publishers Inc., 2nd edition, 2010.
- [Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6) :311–317, June 1975.
- [RDP05] Vincent Ross, Denis Dion, and Guy Potvin. Detailed analytical approach to the gaussian surface bidirectional reflectance distribution function specular component applied to the sea surface. *J. Opt. Soc. Am. A*, 22(11) :2442–2453, Nov 2005.

- [RH01] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of ACM SIGGRAPH' 01*, pages 497–500, 2001.
- [RMS⁺08] Martin Rump, Gero Müller, Ralf Sarlette, Dirk Koch, and Reinhard Klein. Photo-realistic rendering of metallic car paint from image-based measurements. *Computer Graphics Forum*, 27(2), 2008.
- [RTB⁺92] John Rhoades, Greg Turk, Andrew Bell, Andrei State, Ulrich Neumann, and Amitabh Varshney. Real-time procedural textures. In *Proceedings of ACM Symposium on Interactive 3D Graphics*, pages 95–100, 1992.
- [Rus98] Szymon Rusinkiewicz. A new change of variables for efficient BRDF representation. In *Rendering Techniques (Proc. Eurographics Workshop on Rendering)*, June 1998.
- [Sch94] Christophe Schlick. An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum*, 13 :233–246, 1994.
- [Sch01] Andreas Schilling. Antialiasing of environment maps. *Computer Graphics Forum*, 20 :5–11, 2001.
- [SHUD10] Stefan Suwelack, Eric Heitz, Roland Unterhinninghofen, and Rüdiger Dillmann. Adaptive GPU Ray Casting Based on Spectral Analysis. In *MIAR 2010 - 5th International Workshop on Medical Imaging and Augmented Reality*, volume 6326 of *Lecture Notes in Computer Science*, pages 169–178, Beijing, China, September 2010. Springer-Verlag.
- [Sim09] Lionel Simonot. Photometric model of diffuse surfaces described as a distribution of interfaced lambertian facets. *Appl. Opt.*, 48(30) :5793–5801, Oct 2009.
- [Smi67] B. Smith. Geometrical shadowing of a random rough surface. *IEEE Trans. on Antennas and Propagation*, 15 :668–671, 1967.
- [SST89] T. Saito, M. Shinya, and T. Takahashi. Highlighting rounded edges. In RaeA. Earnshaw and Brian Wyvill, editors, *New Advances in Computer Graphics*, pages 613–629. Springer Japan, 1989.
- [Sta99] Jos Stam. Diffraction shaders. In *Proceedings of ACM SIGGRAPH' 99*, pages 101–110, 1999.
- [Sta01] Jos Stam. An illumination model for a skin layer bounded by rough surfaces. In *Rendering Techniques*, pages 39–52, 2001.
- [TLQ⁺05] Ping Tan, Stephen Lin, Long Quan, Baining Guo, and Heung-Yeung Shum. Multiresolution reflectance filtering. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, pages 111–116, 2005.
- [TLQ⁺08] Ping Tan, Stephen Lin, Long Quan, Baining Guo, and Harry Shum. Filtering and rendering of resolution-dependent reflectance models. *IEEE Trans. on Visualization and Computer Graphics*, 14(2) :412–425, March 2008.
- [Tok05] Michael Toksvig. Mipmapping normal maps. *Journal of Graphics Tools*, 10(3) :65–71, 2005.
- [TR75] T. S. Trowbridge and K. P. Reitz. Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am.*, 65(5) :531–536, May 1975.
- [TS67] K.E. Torrance and E.M. Sparrow. Theory for off-specular reflection from roughened surfaces. *J. Opt. Soc. Am.*, 57(9) :1105–1112, Sep 1967.
- [TT91] Toshimitsu Tanaka and Tokiichiro Takahashi. Scientific visualization of physical phenomena. chapter *Precise Rendering Method for Edge Highlighting*, pages 283–298. Springer-Verlag New York, Inc., New York, NY, USA, 1991.
- [Vea98] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford, CA, USA, 1998.
- [vGSK98] Bram van Ginneken, Marigo Stavridi, and Jan J. Koenderink. Diffuse and specular reflectance from rough surfaces. *Appl. Opt.*, 37(1) :130–139, Jan 1998.

- [Wag67] R. J. Wagner. Shadowing of randomly rough surfaces. *The Journal of the Acoustical Society of America*, 41(1) :138–147, 1967.
- [War92] Gregory J. Ward. Measuring and modeling anisotropic reflection. *ACM SIGGRAPH Comput. Graph.*, 26(2) :265–272, July 1992.
- [WDR09] Hongzhi Wu, Julie Dorsey, and Holly Rushmeier. Characteristic point maps. *Computer Graphics Forum*, 28(4) :1227–1236, 2009.
- [WDR11] Hongzhi Wu, Julie Dorsey, and Holly Rushmeier. Physically-based interactive bi-scale material design. *ACM Trans. Graph.*, 30(145), 12/2011 2011.
- [Wil83] Lance Williams. Pyramidal parametrics. In *Proceedings of ACM SIGGRAPH '83*, pages 1–11, 1983.
- [WMLT07] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the Eurographics Symposium on Rendering*, pages 195–206, 2007.
- [Wor02] S. Worley. Advanced antialiasing. In *Texturing and Modeling : A Procedural Approach*, chapter 5, pages 157–176. Morgan Kaufmann, 2002.
- [WRG⁺09] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. In *Proceedings of ACM SIGGRAPH Asia '09*, pages 133 :1–133 :10, 2009.
- [WW07] Andrea Weidlich and Alexander Wilkie. Arbitrarily layered micro-facet surfaces. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pages 171–178, 2007.
- [XSD⁺13] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Trans. Graph.*, 32(6) :209 :1–209 :11, November 2013.
- [YHJ⁺14] Ling-Qi Yan, Miloš Hašan, Wenzel Jakob, Jason Lawrence, Steve Marschner, and Ravi Ramamoorthi. Rendering glints on high-resolution normal-mapped specular surfaces. *Proceedings of ACM SIGGRAPH '14*, 33(4), 2014.
- [Zhu03] Y. Zhu. *Multisensor Decision and Estimation Fusion*. Springer, 2003.

Quatrième partie

Annexes

Annexe A

Dérivation de la fonction de masquage et d'ombrage de Smith

A.1 Dérivation de la fonction de masquage

In this section, we derive $G_1^{\text{dist}}(\omega_o)$ (denoted G_1^{dist} for convenience) starting from Equation (4.18) :

$$\cos \theta_o = G_1^{\text{dist}} \int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m.$$

Slope/Normal Transformations The most complicated step consists of computing the integral, which is defined in the space of the normals. It is more convenient to solve this integral in slope space. We recall that the surface slope associated with a normal $\omega_m = (x_m, y_m, z_m)$ is defined by :

$$\tilde{m}(\omega_m) = (x_{\tilde{m}}, y_{\tilde{m}}) = (-x_m/z_m, -y_m/z_m),$$

and reciprocally :

$$\omega_m(\tilde{m}) = (x_m, y_m, z_m) = \frac{1}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}}(-x_{\tilde{m}}, -y_{\tilde{m}}, 1),$$

and that the slope distribution P^{22} is linked to the distribution of normals by the relationship :¹

$$P^{22}(\tilde{m}) d\tilde{m} = (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m.$$

By using this change of variable in Equation (4.18), we write :

$$\int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\langle \omega_o, \omega_m(\tilde{m}) \rangle}{\omega_g \cdot \omega_m(\tilde{m})} P^{22}(\tilde{m}) d\tilde{m},$$

where $[-\infty, +\infty]^2$ is the cartesian 2D space where the slopes are defined. Since $\omega_g = (0, 0, 1)$, we get :

$$\omega_g \cdot \omega_m(\tilde{m}) = \frac{1}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}},$$

1. The Jacobian of the normal to slope transformation is $\left\| \frac{\partial \omega_m}{\partial \tilde{m}} \right\| = |\omega_m \cdot \omega_g|^3$ and we use it to derive the slope distribution $P^{22}(\tilde{m}) = |\omega_m \cdot \omega_g|^4 D(\omega_m)$.

The clamped dot product can be expanded as :

$$\langle \omega_o, \omega_m(\tilde{m}) \rangle = \frac{\chi^+(-x_o x_{\tilde{m}} - y_o y_{\tilde{m}} + z_o)(-x_o x_{\tilde{m}} - y_o y_{\tilde{m}} + z_o)}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}},$$

and so the integral becomes :

$$\begin{aligned} & \int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \chi^+(-x_o x_{\tilde{m}} - y_o y_{\tilde{m}} + z_o)(-x_o x_{\tilde{m}} - y_o y_{\tilde{m}} + z_o) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dx_{\tilde{m}} dy_{\tilde{m}}. \end{aligned}$$

Without loss of generality, we can assume that the view direction is aligned to the x -axis (i.e., $\omega_o = (\sin \theta_o, 0, \cos \theta_o)$) :

$$\begin{aligned} & \int_{\Omega} \langle \omega_o, \omega_m \rangle D(\omega_m) d\omega_m \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \chi^+(-\sin \theta_o x_{\tilde{m}} + \cos \theta_o)(-\sin \theta_o x_{\tilde{m}} + \cos \theta_o) P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dx_{\tilde{m}} dy_{\tilde{m}} \\ &= \int_{-\infty}^{+\infty} \chi^+(-\sin \theta_o x_{\tilde{m}} + \cos \theta_o)(-\sin \theta_o x_{\tilde{m}} + \cos \theta_o) \left(\int_{-\infty}^{+\infty} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dy_{\tilde{m}} \right) dx_{\tilde{m}} \\ &= \int_{-\infty}^{+\infty} \chi^+(-\sin \theta_o x_{\tilde{m}} + \cos \theta_o)(-\sin \theta_o x_{\tilde{m}} + \cos \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}}, \end{aligned}$$

where $P^{2-}(x_{\tilde{m}}) = \int_{-\infty}^{+\infty} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dy_{\tilde{m}}$ is the 1D slope distribution in the view direction (aligned with the x -axis). Since :

$$-\sin \theta_o x_{\tilde{m}} + \cos \theta_o > 0 \Rightarrow x_{\tilde{m}} < \cot \theta_o,$$

we can drop the Heaviside function by changing the integration domain :

$$\begin{aligned} & \int_{-\infty}^{+\infty} \chi^+(-\sin \theta_o x_{\tilde{m}} + \cos \theta_o)(-\sin \theta_o x_{\tilde{m}} + \cos \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\ &= \int_{-\infty}^{\cot \theta_o} (-\sin \theta_o x_{\tilde{m}} + \cos \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}}. \end{aligned}$$

Now we can return to Equation (4.18) :

$$\cos \theta_o = G_1^{\text{dist}} \int_{-\infty}^{\cot \theta_o} (-\sin \theta_o x_{\tilde{m}} + \cos \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}}.$$

By dividing by $\sin \theta_o$ on both sides, we get :

$$\cot \theta_o = G_1^{\text{dist}} \int_{-\infty}^{\cot \theta_o} (-x_{\tilde{m}} + \cot \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}}.$$

Since microfacet distributions are centered, the average slope in any direction is zero ($\int_{-\infty}^{+\infty} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} = 0$) and we can introduce this term in the equation :

$$\cot \theta_o = G_1^{\text{dist}} \int_{-\infty}^{+\infty} x_{\tilde{m}} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} + G_1^{\text{dist}} \int_{-\infty}^{\cot \theta_o} (-x_{\tilde{m}} + \cot \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}},$$

and by using $\cot \theta_o = (1 - G_1^{\text{dist}}) \cot \theta_o + G_1^{\text{dist}} \cot \theta_o$:

$$\begin{aligned} (1 - G_1^{\text{dist}}) \cot \theta_o + G_1^{\text{dist}} \cot \theta_o &= G_1^{\text{dist}} \int_{-\infty}^{+\infty} x_{\tilde{m}} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} + G_1^{\text{dist}} \int_{-\infty}^{\cot \theta_o} (-x_{\tilde{m}} + \cot \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\ (1 - G_1^{\text{dist}}) \cot \theta_o &= G_1^{\text{dist}} \int_{-\infty}^{+\infty} x_{\tilde{m}} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} + G_1^{\text{dist}} \int_{-\infty}^{\cot \theta_o} (-x_{\tilde{m}} + \cot \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} - G_1^{\text{dist}} \cot \theta_o \end{aligned}$$

and since P^{2-} integrates to 1 we have $G_1^{\text{dist}} \cot \theta_o = G_1^{\text{dist}} \int_{-\infty}^{+\infty} \cot \theta_o P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}}$:

$$\begin{aligned} (1 - G_1^{\text{dist}}) \cot \theta_o &= G_1^{\text{dist}} \int_{-\infty}^{+\infty} x_{\tilde{m}} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} + G_1^{\text{dist}} \int_{-\infty}^{\cot \theta_o} (-x_{\tilde{m}} + \cot \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\ &\quad - G_1^{\text{dist}} \int_{-\infty}^{+\infty} \cot \theta_o P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\ &= G_1^{\text{dist}} \left(\int_{-\infty}^{+\infty} x_{\tilde{m}} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} - \int_{-\infty}^{\cot \theta_o} x_{\tilde{m}} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \right) \\ &\quad + G_1^{\text{dist}} \left(\int_{-\infty}^{\cot \theta_o} \cot \theta_o P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} - \int_{-\infty}^{+\infty} \cot \theta_o P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \right) \\ &= G_1^{\text{dist}} \int_{\cot \theta_o}^{+\infty} x_{\tilde{m}} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} - G_1^{\text{dist}} \int_{\cot \theta_o}^{+\infty} \cot \theta_o P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\ &= G_1^{\text{dist}} \int_{\cot \theta_o}^{\infty} (x_{\tilde{m}} - \cot \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}}. \end{aligned}$$

By dividing by G_1^{dist} on each side, we get :

$$\frac{(1 - G_1^{\text{dist}})}{G_1^{\text{dist}}} = \frac{1}{\cot \theta_o} \int_{\cot \theta_o}^{\infty} (x_{\tilde{m}} - \cot \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}},$$

which leads to the final form :

$$\boxed{G_1^{\text{dist}}(\omega_o) = \frac{1}{1 + \Lambda(\omega_o)},}$$

where function Λ is defined by :

$$\boxed{\Lambda(\omega_o) = \frac{1}{\cot \theta_o} \int_{\cot \theta_o}^{\infty} (x_{\tilde{m}} - \cot \theta_o) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}}.}$$

Our derivation, based on the projected area, has lead us to the generalized form of the Smith masking term [Bro80, WMLT07].

A.2 Dérivation de la fonction de masquage et d'ombrage corrélée par les hauteurs

In this section, we recall the derivation of the height-correlated form of the joint masking-shadowing function [RDP05, HNPN13a, DHI⁺13] presented in Equation (3.7) :

$$G_2(\omega_o, \omega_i, \omega_m) = \frac{\chi^+(\omega_o \cdot \omega_m) \chi^+(\omega_i \cdot \omega_m)}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)}.$$

The microsurface is defined by the distribution of normals $D(\omega_m)$, and the associated slope distribution is $P^{22}(\tilde{m})$ as presented in Appendix A. We introduce $P^1(h)$, the height distribution of the microsurface. Note that the slopes of the microsurface are simply the gradients of the heights : $\tilde{m} = \nabla h$. Smith's derivation [Smi67, WMLT07] gives the probability that a point at height h with non-backfacing normal ω_m is visible from direction ω_o :

$$G_1(\omega_o, \omega_m, h) = G_1^{\text{local}}(\omega_o, \omega_m) G_1^{\text{dist}}(\omega_o, h),$$

where the local and distant masking functions, presented in Chapter 4, are given by :

$$\begin{aligned} G_1^{\text{local}}(\omega_o, \omega_m) &= \chi^+(\omega_o \cdot \omega_m), \\ G_1^{\text{dist}}(\omega_o, h) &= \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o)}. \end{aligned}$$

The height-averaged form is given by :

$$\begin{aligned} G_1(\omega_o, \omega_m) &= G_1^{\text{local}}(\omega_o, \omega_m) \int_{-\infty}^{+\infty} G_1^{\text{dist}}(\omega_o, h) P^1(h) dh \\ &= \chi^+(\omega_o \cdot \omega_m) \int_{-\infty}^{+\infty} \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o)} P^1(h) dh \\ &= \frac{\chi^+(\omega_o \cdot \omega_m)}{1 + \Lambda(\omega_o)}. \end{aligned}$$

which is the Smith masking function from Equation (4.20). Now, if we suppose that there is no directional correlation for masking from directions ω_o and ω_i , then the probability that a point at height h is visible from both directions is just the product of the probabilities :

$$\begin{aligned} G_2(\omega_o, \omega_i, \omega_m, h) &= G_1(\omega_o, \omega_m, h) G_1(\omega_i, \omega_m, h) \\ &= G_1^{\text{local}}(\omega_o, \omega_m) G_1^{\text{dist}}(\omega_o, h) G_1^{\text{local}}(\omega_i, \omega_m) G_1^{\text{dist}}(\omega_i, h) \\ &= \chi^+(\omega_o \cdot \omega_m) \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o)} \chi^+(\omega_i \cdot \omega_m) \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_i)} \\ &= \chi^+(\omega_o \cdot \omega_m) \chi^+(\omega_i \cdot \omega_m) \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)}, \end{aligned}$$

and the height-averaged form is given by :

$$\begin{aligned}
G_2(\omega_o, \omega_i, \omega_m) &= \int_{-\infty}^{+\infty} G_2(\omega_o, \omega_i, \omega_m) P^1(h) dh \\
&= \int_{-\infty}^{+\infty} \chi^+(\omega_o \cdot \omega_m) \chi^+(\omega_i \cdot \omega_m) \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)} P^1(h) dh \\
&= \chi^+(\omega_o \cdot \omega_m) \chi^+(\omega_i \cdot \omega_m) \int_{-\infty}^{+\infty} \left(\int_{-\infty}^h P^1(h') dh' \right)^{\Lambda(\omega_o) + \Lambda(\omega_i)} P^1(h) dh \\
&= \frac{\chi^+(\omega_o \cdot \omega_m) \chi^+(\omega_i \cdot \omega_m)}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)},
\end{aligned}$$

which is the height-correlated masking-shadowing function presented in Equation (3.7).

A.3 Code MATLAB pour le Weak White Furnace Test

In this section, we provide code to numerically compute the integral in Equation (4.15) :

$$\int_{\Omega_i} \frac{G_1(\omega_o, \omega_h) D(\omega_h)}{4 |\omega_g \cdot \omega_o|} d\omega_i = 1,$$

with Beckmann and GGX distributions and their associated Smith masking functions.

```
function [integral] = TEST_BECKMANN(alpha, theta_o)

% view vector
V = [sin(theta_o) 0 cos(theta_o)];
% masking (rational approximation for Lambda)
a = 1 / (alpha * tan(theta_o));
if a < 1.6
    Lambda = (1 - 1.259*a + 0.396*a^2) / (3.535*a + 2.181*a^2);
else
    Lambda = 0;
end
G = 1 / (1 + Lambda);

integral = 0;
dtheta = 0.05;
dphi = 0.05;
for theta = 0:dtheta:pi
for phi = 0:dphi:2*pi
    % reflected vector
    L = [cos(phi)*sin(theta) sin(phi)*sin(theta) cos(theta)];
    % half vector
    H = (V + L) / norm(V + L);
    % Beckmann distribution
    if H(3) > 0
        % angle associated with H
        theta_h = acos(H(3));
        D = exp(-(tan(theta_h)/alpha)^2) / (pi * alpha^2 * H(3)^4);
    else
        continue;
    end
    % integrate
    integral = integral + sin(theta) * D * G / abs(4 * V(3));
end
end

% display integral (should be 1)
integral = integral * dphi * dtheta;
end
```

```

function [integral] = TEST_BECKMANN_ANISO(alpha_x, alpha_y, theta_o, phi_o)

% view vector
V = [cos(phi_o)*sin(theta_o) sin(phi_o)*sin(theta_o) cos(theta_o)];
% alpha in view direction
alpha_o = sqrt(cos(phi_o)^2*alpha_x^2 + sin(phi_o)^2*alpha_y^2);
% masking (rational approximation for Lambda)
a = 1 / (alpha_o * tan(theta_o));
if a < 1.6
    Lambda = (1 - 1.259*a + 0.396*a^2) / (3.535*a + 2.181*a^2);
else
    Lambda = 0;
end
G = 1 / (1 + Lambda);

integral = 0;
dtheta = 0.05;
dphi = 0.05;
for theta = 0:dtheta:pi
for phi = 0:dphi:2*pi
    % reflected vector
    L = [cos(phi)*sin(theta) sin(phi)*sin(theta) cos(theta)];
    % half vector
    H = (V + L) / norm(V + L);
    % Beckmann distribution
    if H(3) > 0
        % slope associated with H
        slope = [-H(1)/H(3) -H(2)/H(3)];
        D = exp(-(slope(1)/alpha_x)^2 - (slope(2)/alpha_y)^2);
        D = D / (pi * alpha_x * alpha_y * H(3)^4);
    else
        continue;
    end
    % integrate
    integral = integral + sin(theta) * D * G / abs(4 * V(3));
end
end

% display integral (should be 1)
integral = integral * dphi * dtheta;
end

```

```

function [integral] = TEST_GGX(alpha, theta_o)

% view vector
V = [sin(theta_o) 0 cos(theta_o)];
% masking
a = 1 / (alpha * tan(theta_o));
Lambda = (-1 + sqrt(1 + 1/a^2)) / 2;
G = 1 / (1 + Lambda);

integral = 0;
dtheta = 0.05;
dphi = 0.05;
for theta = 0:dtheta:pi
for phi = 0:dphi:2*pi
% reflected vector
L = [cos(phi)*sin(theta) sin(phi)*sin(theta) cos(theta)];
% half vector
H = (V + L) / norm(V + L);
% GGX distribution
if H(3) > 0
% angle associated with H
theta_h = acos(H(3));
D = 1 / (1 + (tan(theta_h)/alpha)^2)^2;
D = D / (pi * alpha^2 * H(3)^4);
else
D = 0;
end
% integrate
integral = integral + sin(theta) * D * G / abs(4 * V(3));
end
end

% display integral (should be 1)
integral = integral * dphi * dtheta;
end

```

```

function [integral] = TEST_GGX_ANISO(alpha_x, alpha_y, theta_o, phi_o)

% view vector
V = [cos(phi_o)*sin(theta_o) sin(phi_o)*sin(theta_o) cos(theta_o)];
% alpha in view direction
alpha_o = sqrt(cos(phi_o)^2*alpha_x^2 + sin(phi_o)^2*alpha_y^2);
% masking
a = 1 / (alpha_o * tan(theta_o));
Lambda = (-1 + sqrt(1 + 1/a^2)) / 2;
G = 1 / (1 + Lambda);

integral = 0;
dtheta = 0.05;
dphi = 0.05;
for theta = 0:dtheta:pi
for phi = 0:dphi:2*pi
% reflected vector
L = [cos(phi)*sin(theta) sin(phi)*sin(theta) cos(theta)];
% half vector
H = (V + L) / norm(V + L);
% GGX distribution
if H(3) > 0
% slope associated with H
slope = [-H(1)/H(3) -H(2)/H(3)];
D = 1/(1 + (slope(1)/alpha_x)^2 + (slope(2)/alpha_y)^2)^2;
D = D / (pi * alpha_x * alpha_y * H(3)^4);
else
D = 0;
end
% integrate
integral = integral + sin(theta) * D * G / abs(4 * V(3));
end
end

% display integral (should be 1)
integral = integral * dphi * dtheta;
end

```

Warning! The values `dtheta` and `dphi` used to discretize the BRDF in the numerical integration are hardcoded. In practice, setting them to 0.05 works well for `alpha` \geq 0.2. If `alpha` is smaller than 0.2 then `dtheta` and `dphi` must be set to smaller values as well, in order to correctly capture the sharp BRDF lobe.

Annexe B

Implémentation MATLAB du facteur r_0 (chapitre 6)

Les notations utilisées dans le code et les numéros des équations correspondent à l'article [HBP13].

```
function [P_zeta_zetaB] = P_zeta_zetaB(zeta_0, tau_B, zeta_B, mu_A, phi)
% Output
% P_zetaA_zetaB: Conditional height CDF, Eq. (21)

% Input
% zeta_0: height of starting point
% tau_B: abscissa along ray B
% zeta_B: height below ray B
% mu_A: slope of ray A
% phi: azimuthal difference between A and B

% Distance to point below A
d = tau_B * tan(phi);
% Abscissa along A
tau_A = tau_B / cos(phi);
% Height on A
zeta_A = zeta_0 + tau_A * mu_A;

% Conditional mean below A, Eq. (19)
mean_zeta_A = zeta_B / (1 + d^2);
% Conditional variance below A, Eq. (20)
var_zeta_A = d^2 / (1 + d^2);

% Conditional height CDF, Eq. (21)
P_zeta_zetaB = 1.0 - 0.5 * erfc( (zeta_A-mean_zeta_A) / sqrt(2*var_zeta_A) );

endfunction
```

```

function [r0] = r0_approx(h_0, nu_A, nu_B, phi)
% Output
% r0: azimuthal correction coefficient (approximation)

% Input
% h0: normalized height
% nu_A: normalized slope of ray A
% nu_B: normalized slope of ray B
% phi: azimuthal difference between A and B

if phi >= pi/2 % No correlation
    r0 = 1;
    return;
endif

a = 0.17/abs(nu_B-nu_A)^10.49;
b = 8.85;

r0 = log(1 + a*phi^b) / log(1 + a*(pi/2)^b);

endfunction

```

```

function [r0] = r0_numerical(h_0, nu_A, nu_B, phi)
% Output
% r0: azimuthal correction coefficient (numerical evaluation)

% Input
% h0: normalized height
% nu_A: normalized slope of ray A
% nu_B: normalized slope of ray B
% phi: azimuthal difference between A and B

% Compute unnormalized parameters
zeta_0 = sqrt(2) * h_0; % Unnormalized height (sigma_zeta=1)
mu_A = sqrt(2) * min(nu_A, nu_B); % Unnormalized slope (sigma_gamma=1)
mu_B = sqrt(2) * max(nu_A, nu_B); % Unnormalized slope (sigma_gamma=1)

% Extreme cases
if phi == 0 % Bistatic 1D
    r0 = 0;
    return;
endif
if phi >= pi/2 % No correlation
    r0 = 1;
    return;
endif

% Computation of r0, Eq. (27)
r0_numerator = 0; % First integral along tau_B, Eq. (27)
r0_denominator = 0; % Second integral along tau_B, Eq. (27)

% Numerical integration loop
% (stop when zeta_B > 4.0)
dtau_B = 0.01;
for tau_B = 0.01 : dtau_B : (4.0-zeta_0)/mu_B

    zeta_B = zeta_0 + tau_B*mu_B; % Height on B
    p_zeta = exp(-0.5*zeta_B^2) / sqrt(2*pi); % Height PDF
    P_zeta = 1.0 - 0.5 * erfc( zeta_B/sqrt(2) ); % Height CDF

    S_A_nB = P_zeta_zetaB(zeta_0, tau_B, zeta_B, mu_A, phi); % Eq. (17)

    % Numerical integration of S_A_B, Eq. (18)
    % (start when zeta_B_ > -4.0)
    S_A_B = 0;
    dzeta_B_ = 0.01;
    for zeta_B_ = -4.0 : dzeta_B_ : zeta_0+tau_B*mu_B
        p_zeta_ = exp(-0.5*zeta_B_^2) / sqrt(2*pi);
        S_A_B = S_A_B + dzeta_B_ * p_zeta_ * P_zeta_zetaB(zeta_0, tau_B, zeta_B_, mu_A, phi);
    end

    % Update integrals
    r0_numerator = r0_numerator + dtau_B * p_zeta * S_A_nB / S_A_B; % Eq. (27)
    r0_denominator = r0_denominator + dtau_B * p_zeta / P_zeta; % Eq. (27)
end

% Compute r0
r0 = r0_numerator / r0_denominator; % Eq. (27)

endfunction

```

Annexe C

Dérivations des jacobiens

C.1 Jacobian of the Normal-to-Slope Transformation

Here we derive the Jacobian $\left\| \frac{\partial \tilde{m}}{\partial \omega_m} \right\|$ used in Equation (10) in the paper. We note a slope as $\tilde{m} = (x_{\tilde{m}}, y_{\tilde{m}})^t$ and its relation to the normal ω_m is given by

$$\omega_m = \frac{(-x_{\tilde{m}}, -y_{\tilde{m}}, 1)^t}{\sqrt{1 + x_{\tilde{m}}^2 + y_{\tilde{m}}^2}}.$$

In spherical coordinates (θ, ϕ) , the normal is given by

$$\omega_m = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)^t$$

and the slopes can thus be obtained by

$$\begin{pmatrix} \theta \\ \phi \end{pmatrix} \rightarrow \begin{pmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{pmatrix} = \begin{pmatrix} -\tan \theta \cos \phi \\ -\tan \theta \sin \phi \end{pmatrix}.$$

These coordinates systems are illustrated in Figure C.1.

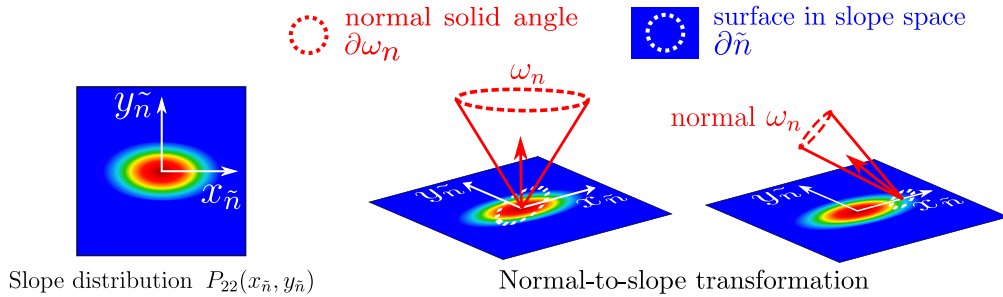


FIGURE C.1 – Normal-to-slope transformation. The Jacobian of this transformation is the ratio between the infinitesimal surface element $\partial \tilde{m}$ and $\partial \omega_m$.

The Jacobian of this transformation corresponds to

$$\left\| \frac{\partial \tilde{m}}{\partial(\theta, \phi)} \right\| = \begin{vmatrix} \frac{\partial x_{\tilde{m}}}{\partial \theta} & \frac{\partial x_{\tilde{m}}}{\partial \phi} \\ \frac{\partial y_{\tilde{m}}}{\partial \theta} & \frac{\partial y_{\tilde{m}}}{\partial \phi} \end{vmatrix} = \begin{vmatrix} -\frac{1}{\cos^2 \theta} \cos \phi & \tan \theta \sin \phi \\ -\frac{1}{\cos^2 \theta} \sin \phi & -\tan \theta \cos \phi \end{vmatrix} = \left| \frac{\sin \theta}{\cos^3 \theta} \right|.$$

The Jacobian of the normal-to-slope transformation is

$$\left\| \frac{\partial \tilde{m}}{\partial \omega_m} \right\| = \left\| \frac{\partial \tilde{m}}{\partial(\theta, \phi)} \right\| \left\| \frac{\partial(\theta, \phi)}{\partial \omega_m} \right\| = \left| \frac{\sin \theta}{\cos^3 \theta} \right| \frac{1}{|\sin \theta|} = \frac{1}{|\cos^3 \theta|}$$

where $\left\| \frac{\partial(\theta, \phi)}{\partial \omega_m} \right\| = \frac{1}{|\sin \theta|}$ is the Jacobian used to transform solid angles into spherical coordinates. Since we define the macronormal $\omega_g = (0, 0, 1)^t$, we have $(\omega_m \cdot \omega_g) = \cos \theta$ and thus

$$\left\| \frac{\partial \tilde{m}}{\partial \omega_m} \right\| = \frac{1}{|\omega_m \cdot \omega_g|^3}. \quad (\text{C.1})$$

C.2 Beckmann Distribution

The Beckmann distribution is built upon a Gaussian distribution of slopes, as illustrated in Figure C.2. The Gaussian slope distribution $P_{22}(\tilde{m})$ is such that

$$1 = \int \int P_{22}(\tilde{m}) d\tilde{m}.$$

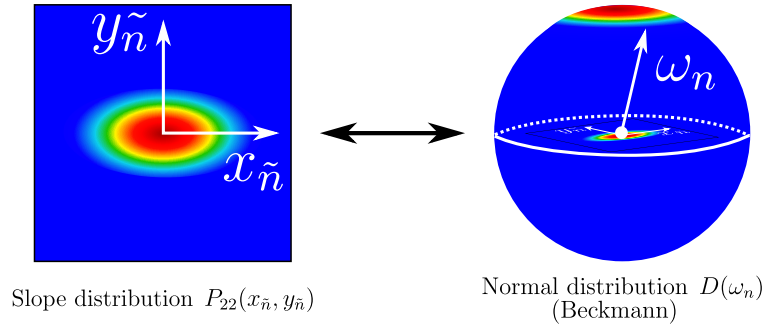


FIGURE C.2 – The Beckmann distribution is built upon a Gaussian slope distribution.

By integrating over the normals instead of over the slopes, with the variable change $d\tilde{m} = \left\| \frac{\partial \tilde{m}}{\partial \omega_m} \right\| d\omega_m$, we get

$$\begin{aligned} 1 &= \int_{\Omega} P_{22}(\tilde{m}) \left\| \frac{\partial \tilde{m}}{\partial \omega_m} \right\| d\omega_m \\ &= \int_{\Omega} \frac{P_{22}(\tilde{m})}{(\omega_m \cdot \omega_g)^3} d\omega_m. \end{aligned}$$

In distribution P_{22} , the density of each microfacet slope is weighted by its projected area $(\omega_m \cdot \omega_g)$ over the macrosurface. In the Beckmann distribution D , each normal is represented independently of any reference plane and is weighted by its real area in world space. This is why it comes with the additional factor $\frac{1}{\omega_m \cdot \omega_g}$

$$\begin{aligned} 1 &= \int_{\Omega} (\omega_m \cdot \omega_g) \frac{P_{22}(\tilde{m})}{(\omega_m \cdot \omega_g)^4} d\omega_m \\ &= \int_{\Omega} (\omega_m \cdot \omega_g) D(\omega_m) d\omega_m \end{aligned}$$

where the Beckmann distribution is

$$D(\omega_m) = \frac{P_{22}(\tilde{m})}{(\omega_m \cdot \omega_g)^4}. \quad (\text{C.2})$$

C.3 Jacobian of the Reflection Operation

Here we derive the Jacobian $\left\| \frac{\partial \omega_i}{\partial \omega_m} \right\|$ used in Sections 4.4 and 5.2 of the paper. The reflection operator is defined by $\omega_i = 2(\omega_m \cdot \omega_o)\omega_m - \omega_o$. To simplify the derivation we choose to center the basis around ω_o instead of ω_m this time

$$\omega_o = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \omega_m = \begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix}$$

and the reflected vector is then

$$\omega_i = 2 \begin{pmatrix} \cos \phi \sin \theta \cos \theta \\ \sin \phi \sin \theta \cos \theta \\ \cos^2 \theta \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

They are illustrated in Figure C.3.

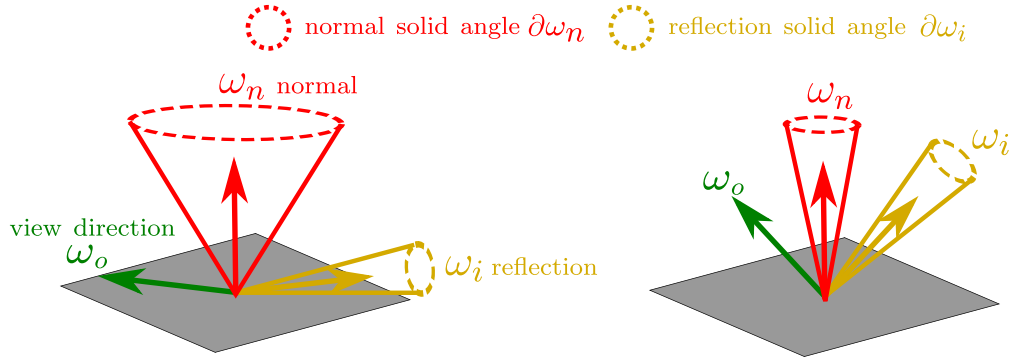


FIGURE C.3 – Reflection operation. The Jacobian of this transformation is the ratio between the infinitesimal solid angles $\partial \omega_i$ and $\partial \omega_m$.

The partial derivatives of ω_i are

$$\frac{\partial \omega_i}{\partial \theta} = 2 \begin{pmatrix} \cos \phi (\cos^2 \theta - \sin^2 \theta) \\ \sin \phi (\cos^2 \theta - \sin^2 \theta) \\ -2 \cos \theta \sin \theta \end{pmatrix} \quad \text{and} \quad \frac{\partial \omega_i}{\partial \phi} = 2 \begin{pmatrix} -\cos \theta \sin \theta \sin \phi \\ \cos \theta \sin \theta \cos \phi \\ 0 \end{pmatrix}$$

and the Jacobian of this transformation is given by the norm of the wedge product

$$\begin{aligned} \left\| \frac{\partial \omega_i}{\partial(\theta, \phi)} \right\| &= \left\| \frac{\partial \omega_i}{\partial \theta} \wedge \frac{\partial \omega_i}{\partial \phi} \right\| = 4 \left\| \begin{array}{ccc} \cos \phi (\cos^2 \theta - \sin^2 \theta) & \sin \phi (\cos^2 \theta - \sin^2 \theta) & -2 \cos \theta \sin \theta \\ -\cos \theta \sin \theta \sin \phi & \cos \theta \sin \theta \cos \phi & 0 \end{array} \right\| \\ &= 4 \sqrt{(2 \cos^2 \theta \sin^2 \theta \cos \phi)^2 + (2 \cos^2 \theta \sin^2 \theta \sin \phi)^2 + (\cos \theta \sin \theta (\cos^2 \theta - \sin^2 \theta))^2} \\ &= 4 |\cos \theta \sin \theta|. \end{aligned}$$

The solid angle transformation of the reflection operation is the Jacobian

$$\left\| \frac{\partial \omega_i}{\partial \omega_m} \right\| = \left\| \frac{\partial \omega_i}{\partial(\theta, \phi)} \right\| \left\| \frac{\partial(\theta, \phi)}{\partial \omega_m} \right\| = 4 |\cos \theta \sin \theta| \frac{1}{|\sin \theta|} = 4 |\cos \theta|$$

and since in this basis $(\omega_m \cdot \omega_o) = \cos \theta$, we get

$$\boxed{\left\| \frac{\partial \omega_i}{\partial \omega_m} \right\| = 4 |\omega_m \cdot \omega_o|}. \quad (\text{C.3})$$

Annexe D

Résultats supplémentaires pour l'intégration en temps-réel (chapitre 10)

D.1 Normal Mapping vs. Displacement Mapping

In this section we compare appearances produced by normal mapping and displacement mapping. Both are computed by applying a noise function with controllable Gaussian statistics on a sphere. The roughness statistics are given by slope standard deviations σ_θ and σ_ϕ in local basis $(\omega_\theta, \omega_\phi)$ defined at each point over a sphere, as shown in Figure D.1 and Table D.1.

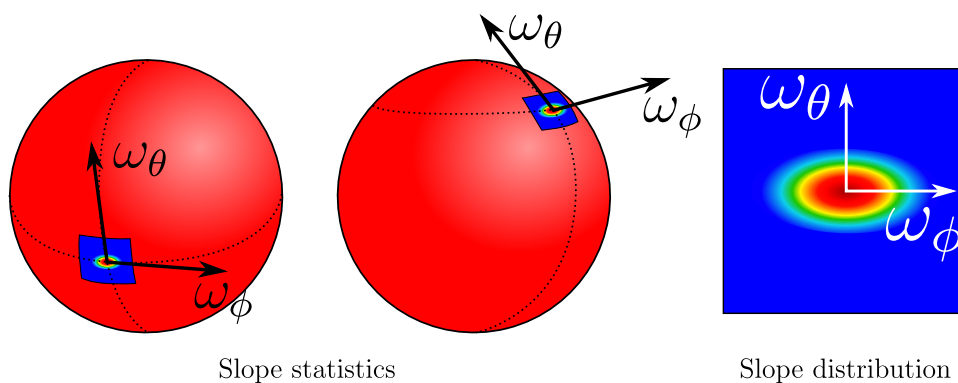


FIGURE D.1 – We use a spherical anisotropic noise primitive that produces Gaussian slope statistics $(\sigma_\theta, \sigma_\phi)$ at any local frame $(\omega_\theta, \omega_\phi)$ on the sphere. The noise statistics are constant in world space and are not affected by the spherical parameterization.

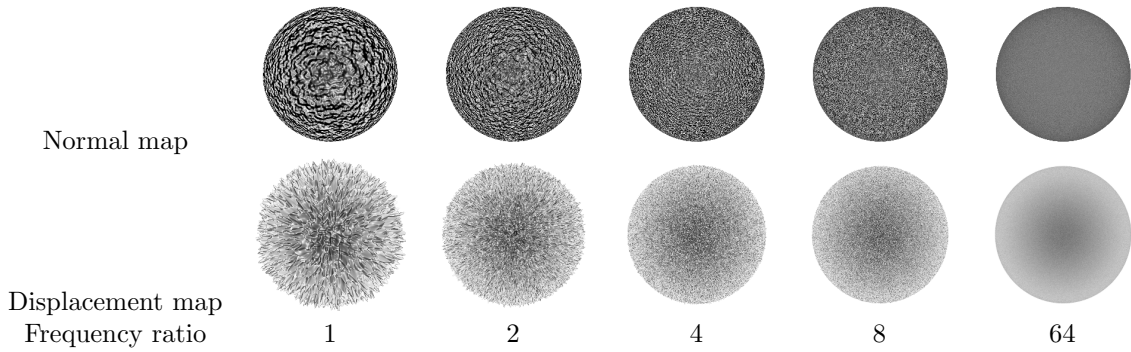


TABLE D.1 – We create procedural normals and procedural displacements with a controllable noise producing constant Gaussian statistics over the surface of a sphere. The supersampled shading produced by high-frequency normals or displacements converges toward the BRDF of the noisy surface, and tends to be smooth.

In Table D.2 we illustrate the difference at macroscale between normal mapping and displacement mapping. For small roughnesses (e.g., $\sigma < 0.25$), the surface of the sphere is rather smooth, and both mappings produce very similar results. However, as roughness increases, displacements produce more appearance effects due to backface culling and projection weighting. In such situation, faces that point toward the viewer become more visible and the appearances produced by both methods strongly diverge. As a side note, one can also observe the displaced silhouettes resulting with displacement mapping, which do not occur with normal mapping.

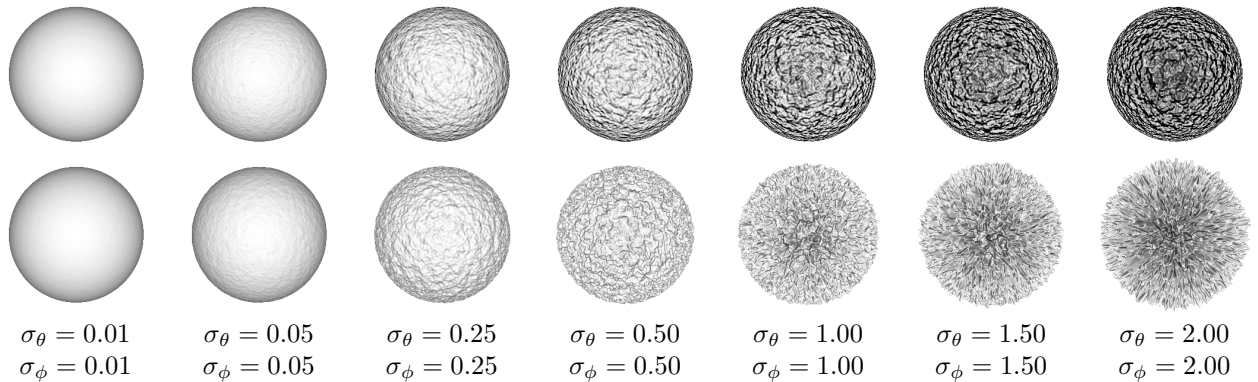


TABLE D.2 – Normal mapping (top row) vs. displacement mapping (middle row) for diffuse microfacets of different roughnesses (bottom row) with one front light, i.e., shining from the camera direction.

D.2 Our Shading Models and Comparisons with Ground Truth

We show the behavior of our BRDF models for different microfacet distributions, lighting configurations, and sampling budgets. We compare results produced at the microscale by normal mapping, displacement mapping, and our model. We show results for diffuse microfacets with a single directional light (Tables D.3 and D.4), with irradiance mapping (Tables D.5 and D.6), and for specular microfacets with environment lighting (Tables D.7 and D.8). We show results for isotropic and anisotropic microfacet distributions, where the anisotropy is latitudinal, i.e., with scratches parallel to the equator, or longitudinal, i.e., with scratches running from the North pole to the South pole. In every table handling anisotropy, the different roughnesses (i.e., for σ_θ vs. σ_ϕ) in each row indicates the “strength” of the anisotropy. The low frequency normal and displacement mapped spheres also give a sense of the orientation and “strength” of the anisotropy.

In the following figures, we denote “Normal map” and “Displacement map” the images computed for low- and high-frequencies (subpixel) normal mapping and displacement mapping, respectively, applied in the same way as in Table D.2. The results denoted under “LEADR $N \times N$ ” are the results computed with our BRDF models with a grid of size $N \times N$ for numerical integration.

Finally, in Table D.9, we illustrate, for a number of roughness values, the tradeoffs between efficiency and bias of LEADR for specular microfacets, according to the grid size of samples.

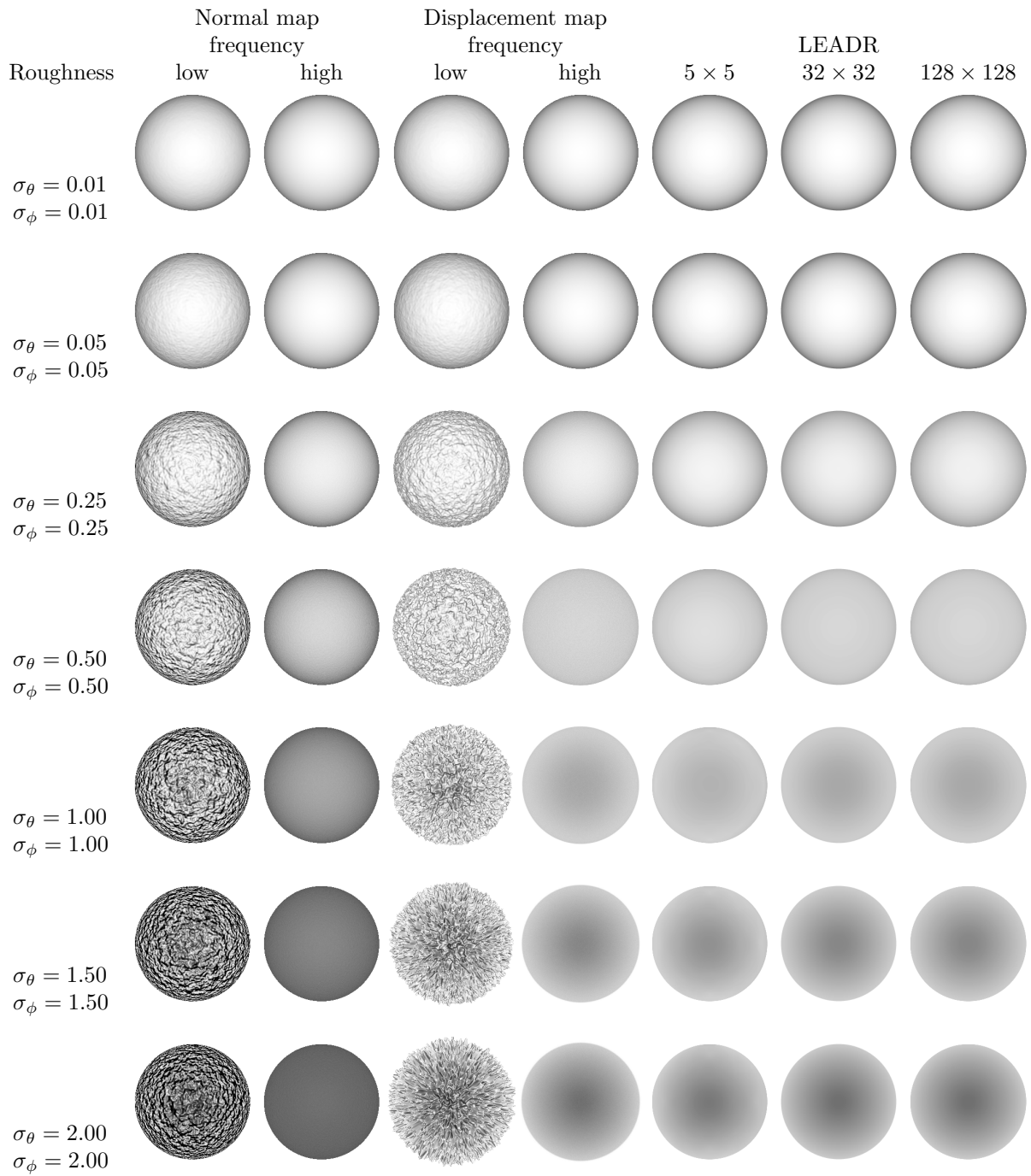


TABLE D.3 – *Isotropic pure diffuse microfacets and one front light.*

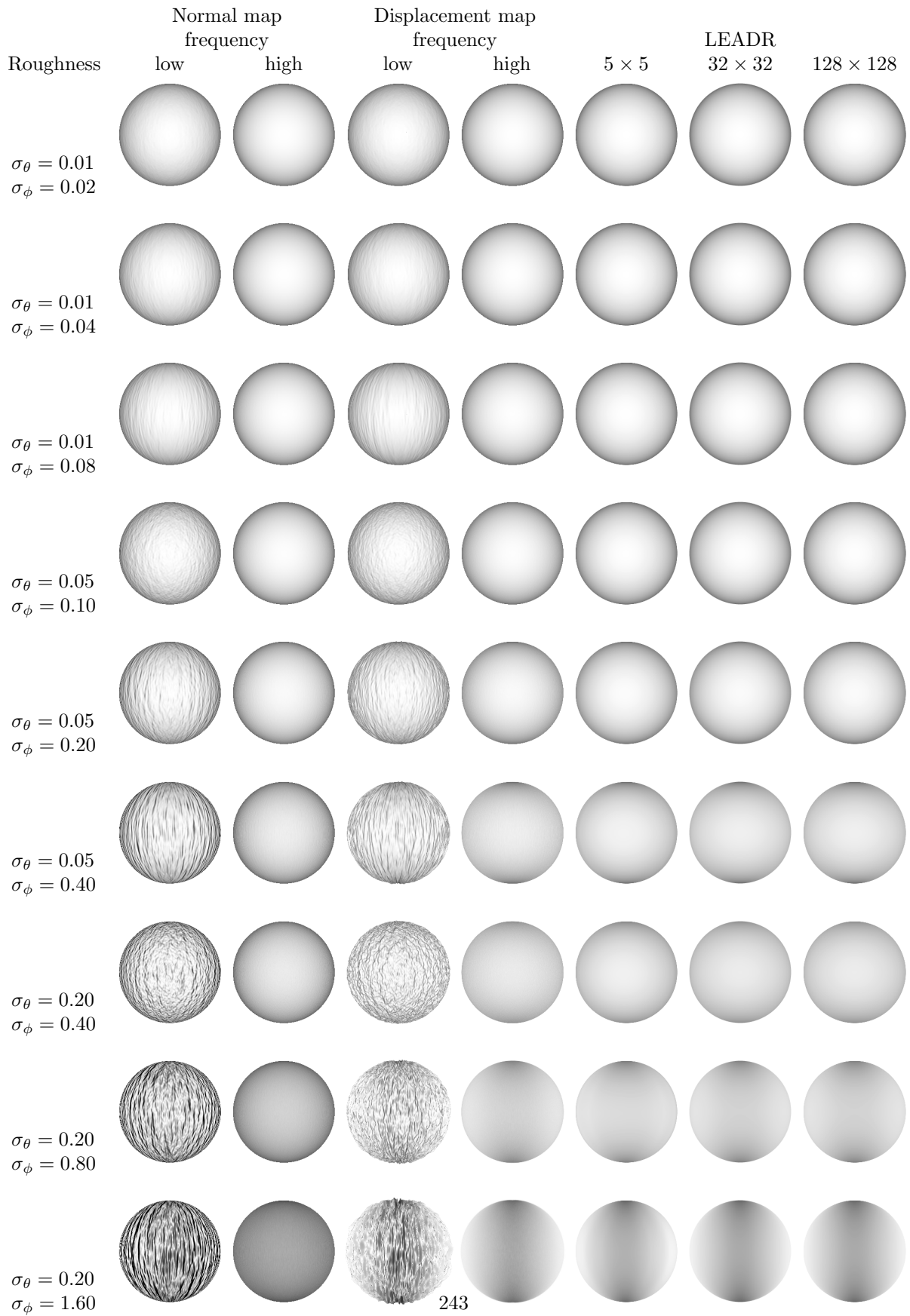


TABLE D.4 – Anisotropic pure diffuse microfacets and one front light.

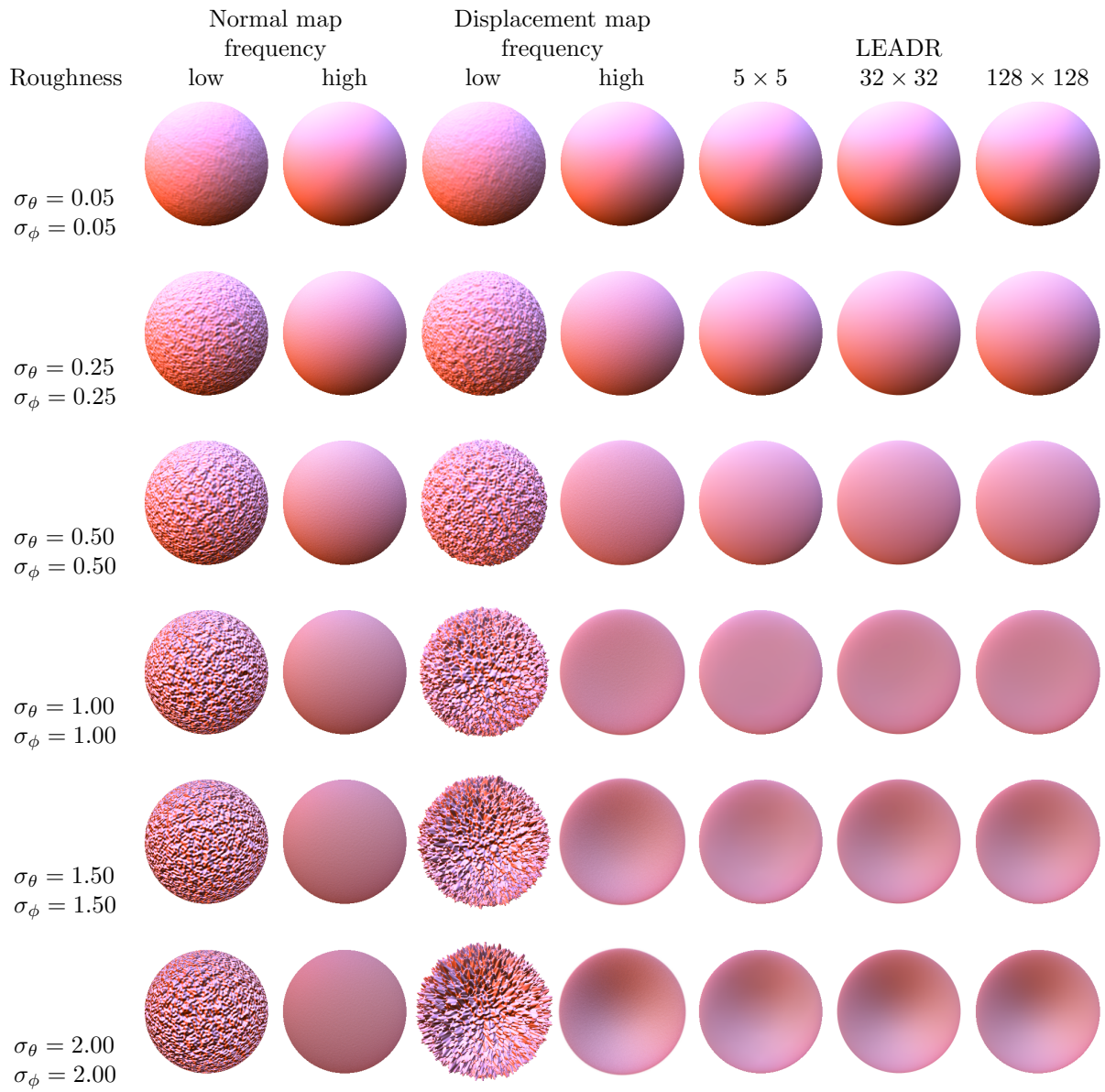


TABLE D.5 – *Isotropic pure diffuse microfacets with an irradiance map.*

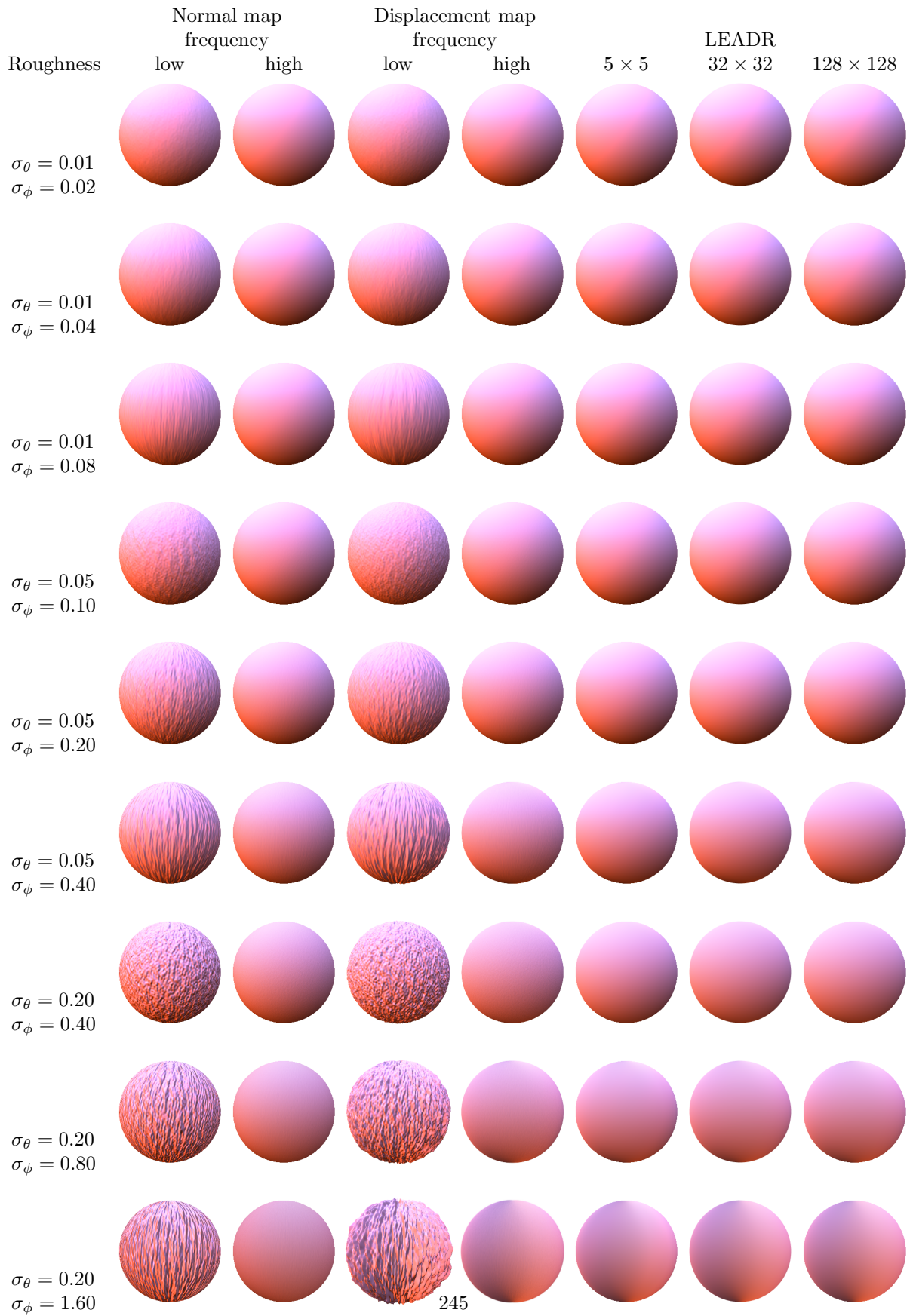


TABLE D.6 – Anisotropic pure diffuse microfacets with an irradiance map.

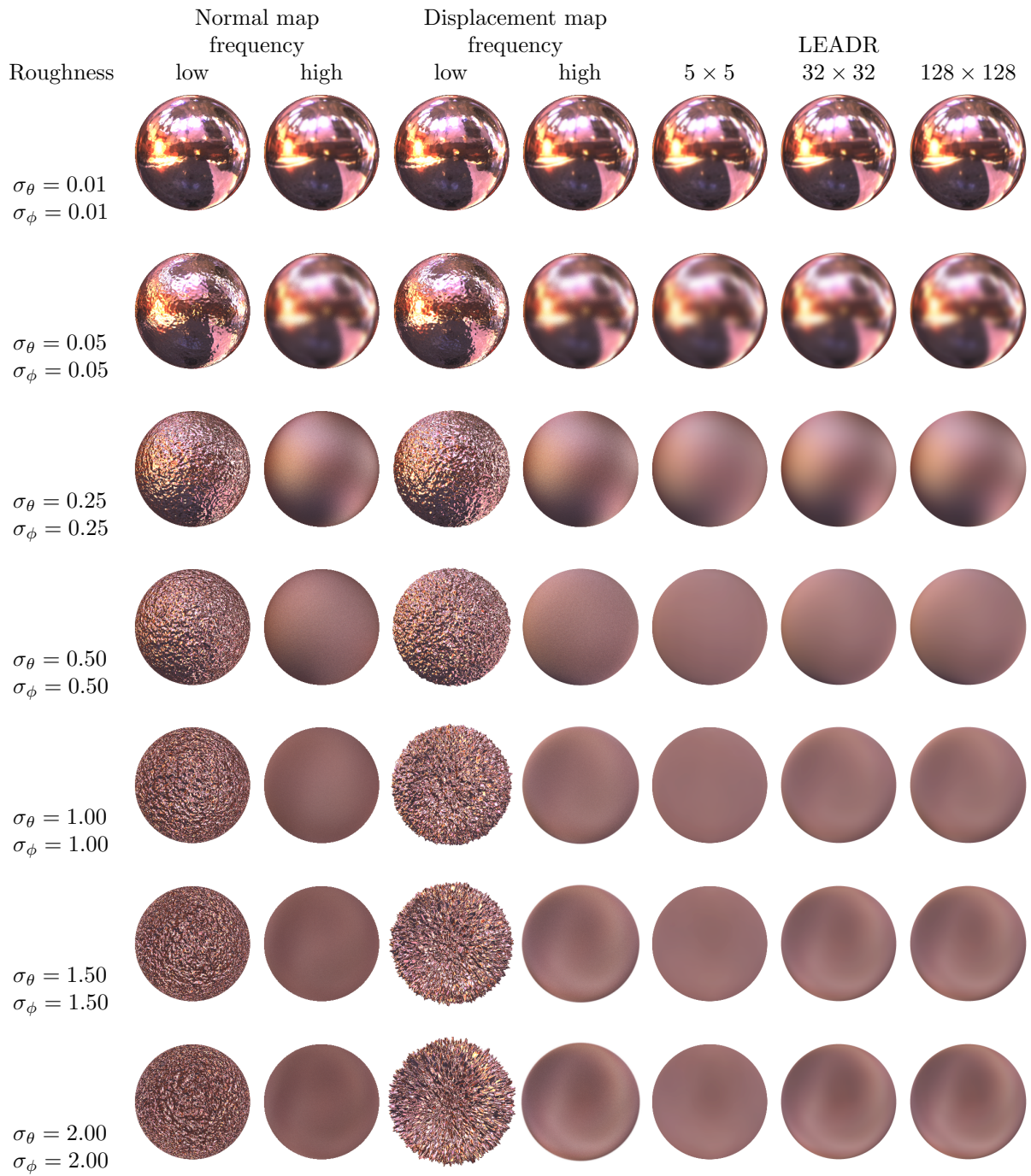


TABLE D.7 – *Isotropic pure specular microfacets with an environment lighting.*

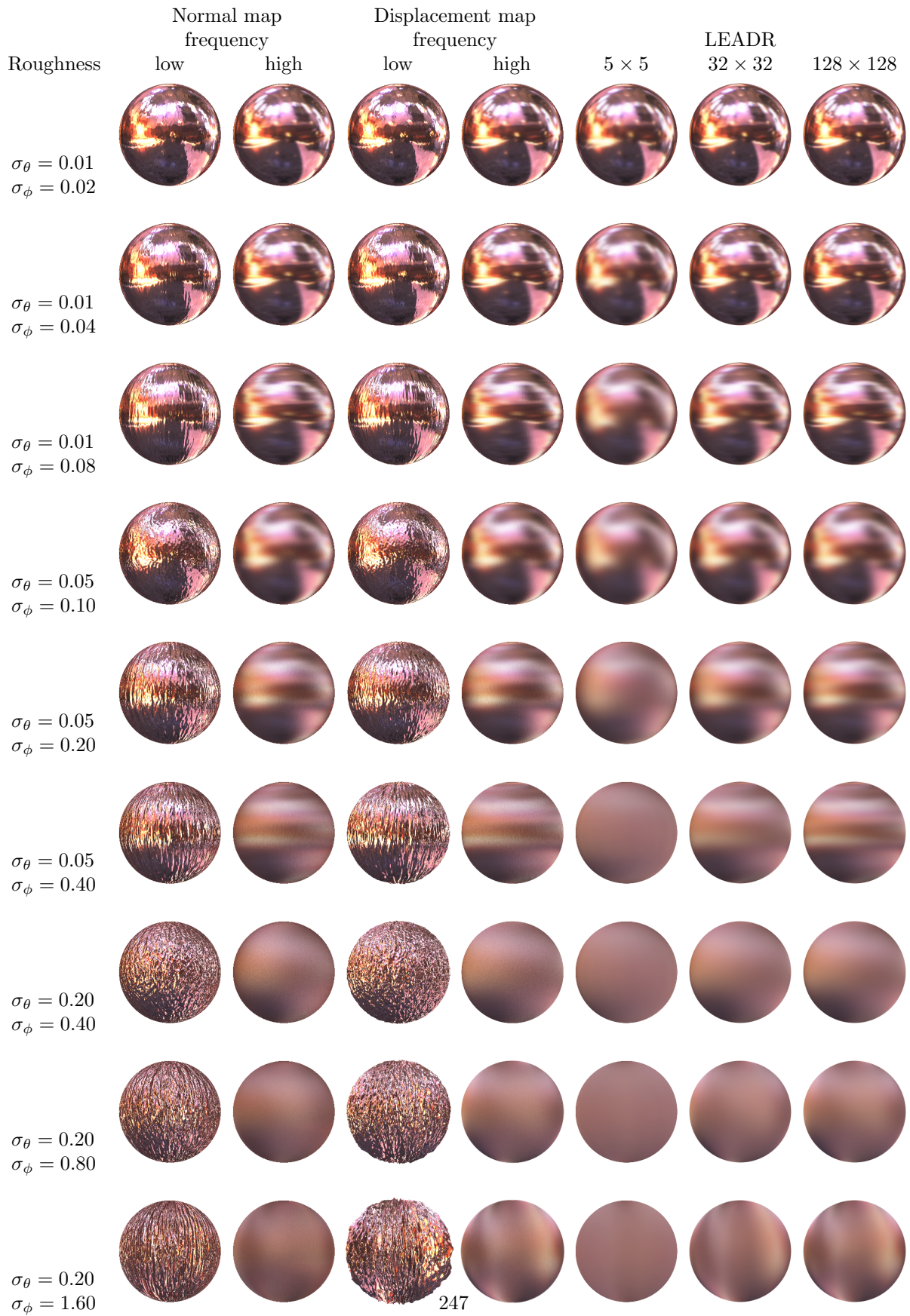


TABLE D.8 – Anisotropic pure specular microfacets with an environment lighting.

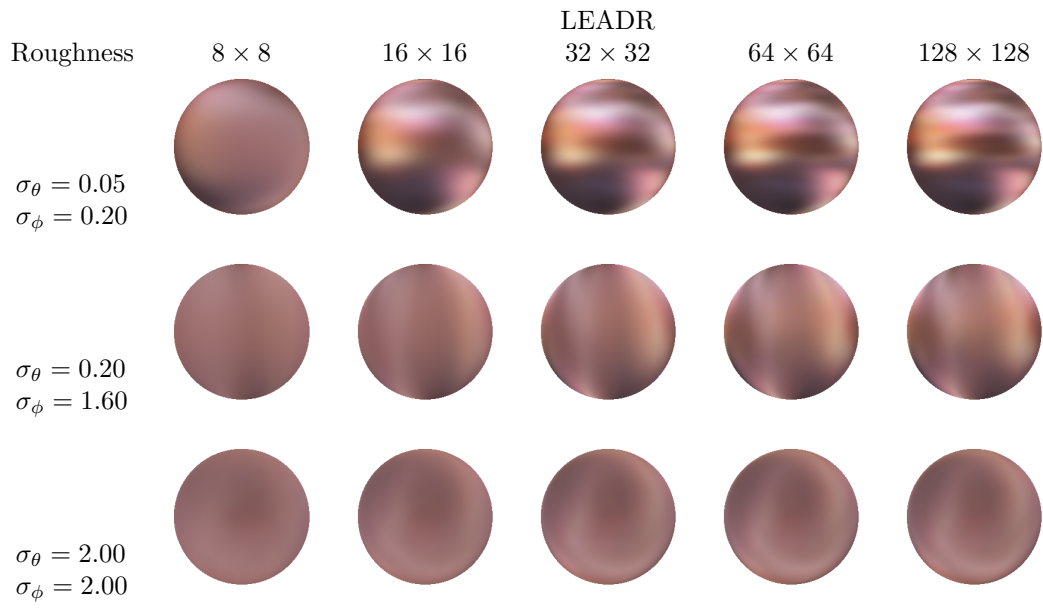


TABLE D.9 – Convergence of LEADR environment map sampling scheme. The tradeoff between efficiency and bias can be adjusted by choosing the number of samples.

Annexe E

Calcul des moments de la distribution de pentes visibles (chapitre 9)

E.1 Gaussian Surface Slope Distribution

In this section we review the definition of a Gaussian microsurface local-orientation/slope generation process, as well as its transformation after applications of rotation operators.

Definition : A Gaussian microsurface local-orientation/slope distribution $\mathcal{N}(\bar{s}, \Sigma)$ expresses the probability with which a microfacet of slope s is present on the microsurface. It is defined by the following *probability distribution function* (PDF) :

$$p_s(s) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(s - \bar{s})^T \Sigma^{-1}(s - \bar{s})\right). \quad (\text{E.1})$$

Expressed about a local orthonormal basis (\vec{x}, \vec{y}) at a microsurface point, the average slope and covariance matrix are :

$$\bar{s} = \begin{pmatrix} \bar{s}_x \\ \bar{s}_y \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} \sigma_x^2 & c_{xy} \\ c_{xy} & \sigma_y^2 \end{pmatrix}.$$

Rotation : The computation of view-dependent slope distributions can be simplified by expressing the distribution in a view-aligned coordinate frame, rotating it to align with $(\vec{o}, \vec{\perp})$, where \vec{o} is the view direction and $\vec{\perp}$ is a direction orthogonal to the view direction. The slope s_o in the view direction and the orthogonal direction s_{\perp} are defined as

$$s_o = s_x \cos \phi_o + s_y \sin \phi_o \quad (\text{E.2})$$

$$s_{\perp} = -s_x \sin \phi_o + s_y \cos \phi_o \quad (\text{E.3})$$

and the inverse transformation is obtained as

$$s_x = s_o \cos \phi_o - s_{\perp} \sin \phi_o \quad (\text{E.4})$$

$$s_y = s_o \sin \phi_o + s_{\perp} \cos \phi_o. \quad (\text{E.5})$$

In order to compute the PDF p_s of the slope distribution *in the rotated view-aligned frame*, we need only compute the new average slope and covariance matrix, taking advantage of the fact that a rotated Gaussian distribution remains a Gaussian distribution :

$$\begin{pmatrix} \bar{s}_o \\ \bar{s}_\perp \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \sigma_o^2 & c_{o\perp} \\ c_{o\perp} & \sigma_\perp^2 \end{pmatrix}$$

The average values are rotated in a similar manner,

$$\bar{s}_o = \bar{s}_x \cos \phi_o + \bar{s}_y \sin \phi_o \quad (\text{E.6})$$

$$\bar{s}_\perp = -\bar{s}_x \sin \phi_o + \bar{s}_y \cos \phi_o, \quad (\text{E.7})$$

and the rotated variances and covariance are given by

$$\begin{aligned} \sigma_o^2 &= \text{var}(s_x \cos \phi_o + s_y \sin \phi_o) \\ &= \sigma_x^2 \cos^2 \phi_o + \sigma_y^2 \sin^2 \phi_o + 2 c_{xy} \cos \phi_o \sin \phi_o \end{aligned} \quad (\text{E.8})$$

$$\begin{aligned} \sigma_\perp^2 &= \text{var}(-s_x \sin \phi_o + s_y \cos \phi_o) \\ &= \sigma_x^2 \sin^2 \phi_o + \sigma_y^2 \cos^2 \phi_o - 2 c_{xy} \cos \phi_o \sin \phi_o \end{aligned} \quad (\text{E.9})$$

$$\begin{aligned} c_{o\perp} &= \text{cov}(s_x \cos \phi_o + s_y \sin \phi_o, -s_x \sin \phi_o + s_y \cos \phi_o) \\ &= (\sigma_y^2 - \sigma_x^2) \cos \phi_o \sin \phi_o + c_{xy} (\cos^2 \phi_o - \sin^2 \phi_o). \end{aligned} \quad (\text{E.10})$$

E.2 Visible Slope Distribution

In the previous section we defined the slope PDF of the microsurface. The visible slope PDF describes the probability that a microfacet with slope s is visible from the view direction $\omega_o(\theta_o, \phi_o)$, and we detail its derivation below.

Definition : The PDF of the visible slope distribution D_s depends on the PDF of the microsurface slope PDF p_s and of the visible (foreshortened) area of the microfacets with respect to the viewing direction. These values depend on the normals of the microfacets. A microfacet with slope $s = (s_x, s_y)$ has normal

$$n = [n_x, n_y, n_z] = \frac{[-s_x, -s_y, 1]^T}{\sqrt{s_x^2 + s_y^2 + 1}}. \quad (\text{E.11})$$

Since the Gaussian slope process is defined on the plane (in 2D) of normal $[0, 0, 1]^T$ the presence of the microfacet (in 3D) has to be weighted by the jacobian of the 2D \rightarrow 3D projection which is $\frac{1}{n_z}$. The foreshortening weight induced by projection towards the viewing direction is the clamped cosine $\max(0, n(s) \cdot \omega_o)$, and so the PDF of the visible slope distribution is

$$D_s(s) = \frac{1}{N} \frac{p_s(s) \max(0, n(s) \cdot \omega_o)}{n_z} \quad (\text{E.12})$$

$$= \frac{1}{N} p_s(s) W(s) H(s), \quad (\text{E.13})$$

where

$$W(s) = -s_x \omega_{ox} - s_y \omega_{oy} + \omega_{oz} \quad (\text{E.14})$$

$$H(s) = \text{Heaviside}(-s_x \omega_{ox} - s_y \omega_{oy} + \omega_{oz}) \quad (\text{E.15})$$

and N is the normalization factor of the distribution

$$N = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p_s(s) W(s) H(s) d^2s. \quad (\text{E.16})$$

Unfortunately, D_s no longer remains a Gaussian distribution after this modification. However, D_s can be well approximated with a Gaussian distribution parameterized by an average vector and covariance matrix. We will detail the derivation of such an approximation below. The starting point of our derivation is given by Ross et al. [?] who derive N for a Gaussian Distribution p_s centered at $(0, 0)$. We use a similar derivation to compute the first and second moments of D_s for an arbitrarily centered p_s , generalizing Ross et al.'s derivation.

Simplification and Derivation : The equations above simplify when expressed in the view-aligned coordinate frame $(\vec{o}, \vec{\perp})$ as

$$W(s) = -s_o \sin \theta_o + \cos \theta_o \quad (\text{E.17})$$

$$H(s) = \text{Heaviside}(-s_o \sin \theta_o + \cos \theta_o) . \quad (\text{E.18})$$

The average, variance and covariance moments of D_s are given by

$$\bar{s}_o(D_s) = \frac{1}{N} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o p_s(s) W(s) H(s) d^2 s \quad (\text{E.19})$$

$$\bar{s}_{\perp}(D_s) = \frac{1}{N} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_{\perp} p_s(s) W(s) H(s) d^2 s \quad (\text{E.20})$$

$$\bar{s}_o^2(D_s) = \frac{1}{N} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o^2 p_s(s) W(s) H(s) d^2 s \quad (\text{E.21})$$

$$\bar{s}_{\perp}^2(D_s) = \frac{1}{N} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_{\perp}^2 p_s(s) W(s) H(s) d^2 s \quad (\text{E.22})$$

$$\bar{s}_{o\perp}(D_s) = \frac{1}{N} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o s_{\perp} p_s(s) W(s) H(s) d^2 s . \quad (\text{E.23})$$

Since $W(s)$ can be decomposed into an sum of two terms (Equation (E.17)), all these integrals can be expressed as sums of two terms as

$$N = -\sin \theta_o I_{s_o} + \cos \theta_o I \quad (\text{E.24})$$

$$\bar{s}_o(D_s) = \frac{1}{N} (-\sin \theta_o I_{s_o^2} + \cos \theta_o I_{s_o}) \quad (\text{E.25})$$

$$\bar{s}_{\perp}(D_s) = \frac{1}{N} (-\sin \theta_o I_{s_o s_{\perp}} + \cos \theta_o I_{s_{\perp}}) \quad (\text{E.26})$$

$$\bar{s}_o^2(D_s) = \frac{1}{N} (-\sin \theta_o I_{s_o^3} + \cos \theta_o I_{s_o^2}) \quad (\text{E.27})$$

$$\bar{s}_{\perp}^2(D_s) = \frac{1}{N} (-\sin \theta_o I_{s_o s_{\perp}^2} + \cos \theta_o I_{s_{\perp}^2}) \quad (\text{E.28})$$

$$\bar{s}_{o\perp}(D_s) = \frac{1}{N} (-\sin \theta_o I_{s_o^2 s_{\perp}} + \cos \theta_o I_{s_o s_{\perp}}) , \quad (\text{E.29})$$

where each of the component integrals are defined as

$$I = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p_s(s) H(s) d^2 s \quad (\text{E.30})$$

$$I_{s_o} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o p_s(s) H(s) d^2 s \quad (\text{E.31})$$

$$I_{s_{\perp}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_{\perp} p_s(s) H(s) d^2 s \quad (\text{E.32})$$

$$I_{s_o^2} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o^2 p_s(s) H(s) d^2 s \quad (\text{E.33})$$

$$I_{s_{\perp}^2} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_{\perp}^2 p_s(s) H(s) d^2 s \quad (\text{E.34})$$

$$I_{s_o s_{\perp}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o s_{\perp} p_s(s) H(s) d^2 s \quad (\text{E.35})$$

$$I_{s_o^2 s_{\perp}} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o^2 s_{\perp} p_s(s) H(s) d^2 s \quad (\text{E.36})$$

$$I_{s_o s_{\perp}^2} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o s_{\perp}^2 p_s(s) H(s) d^2 s \quad (\text{E.37})$$

$$I_{s_o^3} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_o^3 p_s(s) H(s) d^2 s. \quad (\text{E.38})$$

We describe the solution to these integrals in the following section. Given the solution for the moments, we obtain the following covariance matrix entries

$$\sigma_o^2(D_s) = \bar{s}_o^2(D_s) - \bar{s}_o(D_s)^2 \quad (\text{E.39})$$

$$\sigma_{\perp}^2(D_s) = \bar{s}_{\perp}^2(D_s) - \bar{s}_{\perp}(D_s)^2 \quad (\text{E.40})$$

$$c_{o\perp}(D_s) = \bar{s}_{o\perp}(D_s) - \bar{s}_o(D_s)\bar{s}_{\perp}(D_s). \quad (\text{E.41})$$

Finally, the average vector and covariance matrix in the canonical surface-aligned coordinate frame is obtained by applying the inverse rotation operator :

$$\bar{s}_x(D_s) = \bar{s}_o(D_s) \cos \phi_o - \bar{s}_{\perp}(D_s) \sin \phi_o \quad (\text{E.42})$$

$$\bar{s}_y(D_s) = \bar{s}_o(D_s) \sin \phi_o + \bar{s}_{\perp}(D_s) \cos \phi_o \quad (\text{E.43})$$

$$\sigma_x^2(D_s) = \sigma_o^2(D_s) \cos^2 \phi_o + \sigma_{\perp}^2(D_s) \sin^2 \phi_o - 2 c_{o\perp}(D_s) \cos \phi_o \sin \phi_o \quad (\text{E.44})$$

$$\sigma_y^2(D_s) = \sigma_o^2(D_s) \sin^2 \phi_o + \sigma_{\perp}^2(D_s) \cos^2 \phi_o + 2 c_{o\perp}(D_s) \cos \phi_o \sin \phi_o \quad (\text{E.45})$$

$$c_{xy}(D_s) = (\sigma_o^2(D_s) - \sigma_{\perp}^2(D_s)) \cos \phi_o \sin \phi_o + c_{o\perp}(D_s)(\cos^2 \phi_o - \sin^2 \phi_o). \quad (\text{E.46})$$

E.3 Solving for Integrals in Equations (E.30) to (E.38)

Given the PDF

$$p_s(s) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}[s_o - \bar{s}_o, s_{\perp} - \bar{s}_{\perp}]\Sigma^{-1}[s_o - \bar{s}_o, s_{\perp} - \bar{s}_{\perp}]^T\right), \quad (\text{E.47})$$

the inverse of the covariance matrix expressed in the coordinate frame of $(\vec{o}, \vec{\perp})$ is

$$\Sigma^{-1} = \frac{1}{\sigma_o^2\sigma_{\perp}^2 - c_{o\perp}^2} \begin{pmatrix} \sigma_{\perp}^2 & -c_{o\perp} \\ -c_{o\perp} & \sigma_o^2 \end{pmatrix}$$

and by expanding the argument in the exponential in Equation (E.47) we obtain

$$p_s(s) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{a}{2}(s_o - \bar{s}_o)^2 - b(s_o - \bar{s}_o)(s_\perp - \bar{s}_\perp) - \frac{c}{2}(s_\perp - \bar{s}_\perp)^2\right), \quad (\text{E.48})$$

where

$$a = \frac{\sigma_\perp^2}{\sigma_o^2\sigma_\perp^2 - c_{o\perp}^2}, \quad b = -\frac{c_{o\perp}}{\sigma_o^2\sigma_\perp^2 - c_{o\perp}^2}, \quad c = \frac{\sigma_o^2}{\sigma_o^2\sigma_\perp^2 - c_{o\perp}^2}$$

and we have

$$\sigma_o^2 = \frac{c}{ac - b^2}, \quad c_{o\perp} = -\frac{b}{ac - b^2}, \quad \sigma_\perp^2 = \frac{a}{ac - b^2}.$$

From the partial derivative of p_s

$$\frac{\partial p_s(s)}{\partial s_o} = (-a(s_o - \bar{s}_o) - b(s_\perp - \bar{s}_\perp)) p_s(s) \quad (\text{E.49})$$

$$\frac{\partial p_s(s)}{\partial s_\perp} = (-b(s_o - \bar{s}_o) - c(s_\perp - \bar{s}_\perp)) p_s(s) \quad (\text{E.50})$$

we obtain

$$s_o p_s(s) = -c_{o\perp} \frac{\partial p_s(s)}{\partial s_\perp} - \sigma_o^2 \frac{\partial p_s(s)}{\partial s_o} + \bar{s}_o p_s(s) \quad (\text{E.51})$$

$$s_\perp p_s(s) = -\sigma_o^2 \frac{\partial p_s(s)}{\partial s_\perp} + c_{o\perp} \frac{\partial p_s(s)}{\partial s_o} + \bar{s}_\perp p_s(s), \quad (\text{E.52})$$

and after multiplication by $H(s)$ and integration by parts, we obtain

$$s_o p_s(s) H(s) = -c_{o\perp} \frac{\partial p_s(s) H(s)}{\partial s_\perp} + c_{o\perp} p_s(s) \frac{\partial H(s)}{\partial s_\perp} - \sigma_o^2 \frac{\partial p_s(s) H(s)}{\partial s_o} + \sigma_o^2 p_s(s) \frac{\partial H(s)}{\partial s_o} + \bar{s}_o p_s(s) \quad (\text{E.53})$$

and

$$s_\perp p_s(s) H(s) = -\sigma_o^2 \frac{\partial p_s(s) H(s)}{\partial s_\perp} + \sigma_o^2 p_s(s) \frac{\partial H(s)}{\partial s_\perp} + c_{o\perp} \frac{\partial p_s(s) H(s)}{\partial s_o} - c_{o\perp} p_s(s) \frac{\partial H(s)}{\partial s_o} + \bar{s}_\perp p_s(s). \quad (\text{E.54})$$

We substitute the integrands in Equations (E.30)-(E.38) with these results and leverage the following properties of the terms in Equations (E.53) and (E.54) to aid in our analytic integration :

- Since H depends only on s_o ,

$$\frac{\partial H(s)}{\partial s_\perp} = 0; \quad (\text{E.55})$$

- Since p_s tends toward 0 at infinity and H is a binary function,

$$\int_{-\infty}^{+\infty} \frac{\partial p_s(s) H(s)}{\partial s_o} ds_o = 0 \text{ and} \quad (\text{E.56})$$

$$\int_{-\infty}^{+\infty} \frac{\partial p_s(s) H(s)}{\partial s_\perp} ds_\perp = 0; \quad (\text{E.57})$$

- The derivative of a Heaviside function is a Dirac, and so

$$\int_{-\infty}^{+\infty} p_s(s) \frac{\partial H(s)}{\partial s_o} ds_o = -p_s(s_o = \cot \theta_o, s_\perp); \quad (\text{E.58})$$

and by integrating over s_\perp we obtain

$$E_1 = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p_s(s) \frac{\partial H(s)}{\partial s_o} d^2s = -\frac{\sigma_o}{\sqrt{2\pi}} \exp\left(-\frac{(\cot \theta_o - \bar{s}_o)^2}{2\sigma_o^2}\right). \quad (\text{E.59})$$

- The integral below can be simplified by noting, as mentioned above, that H depends only on s_o :

$$E_2 = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p_s(s) H(s) d^2s = 1 - \frac{1}{2} \operatorname{erfc} \left(\frac{\cot \theta_o - \bar{s}_o}{\sqrt{2}\sigma_o} \right) . \quad (\text{E.60})$$

By applying Equations (E.55) to (E.60), we can integrate Equations (E.30)-(E.38) and obtain

$$I = E_2 \quad (\text{E.61})$$

$$I_{s_o} = -E_1 + \bar{s}_o E_2 \quad (\text{E.62})$$

$$I_{s_\perp} = -\frac{c_{o\perp}}{\sigma_o^2} E_1 + \bar{s}_\perp E_2 \quad (\text{E.63})$$

$$I_{s_o^2} = -\cot \theta_o E_1 + \sigma_o^2 E_2 + \bar{s}_o I_{s_o} \quad (\text{E.64})$$

$$I_{s_\perp^2} = -\frac{c_{o\perp}}{\sigma_o^2} (\bar{s}_\perp + \frac{c_{o\perp}}{\sigma_o^2} (\cot \theta_o - \bar{s}_o)) E_1 + \sigma_\perp^2 E_2 + \bar{s}_\perp I_{s_\perp} \quad (\text{E.65})$$

$$I_{s_o s_\perp} = -\frac{c_{o\perp}}{\sigma_o^2} \cot \theta_o E_1 + c_{o\perp} E_2 + \bar{s}_\perp I_{s_o} \quad (\text{E.66})$$

$$I_{s_o^2 s_\perp} = -\frac{c_{o\perp}}{\sigma_o^2} \cot^2 \theta_o E_1 + 2 c_{o\perp} I_{s_o} + \bar{s}_\perp I_{s_o^2} \quad (\text{E.67})$$

$$I_{s_o s_\perp^2} = -\left(\frac{\sigma_o^2 \sigma_\perp^2 - c_{o\perp}^2}{\sigma_o^2} + (\bar{s}_\perp + \frac{c_{o\perp}}{\sigma_o^2} (\cot \theta_o - \bar{s}_o))^2 \right) E_1 + \bar{s}_o I_{s_\perp^2} + 2 c_{o\perp} I_{s_\perp} \quad (\text{E.68})$$

$$I_{s_o^3} = 2 \sigma_o^2 I_{s_o} - \cot^2 \theta_o E_1 + \bar{s}_o I_{s_o^2} . \quad (\text{E.69})$$

E.4 Implementation and Pseudocode

We propose an example pseudocode implementation (see following page) to compute the parameters of D_s given a p_s distribution and (distant) point of view (θ, ϕ) as input.

```

void D_s(
// input: surface slope distribution p_s
double ps_sx0, double ps_sy0, double ps_varx, double ps_vary, double ps_cxy,
// input: view direction
double theta, double phi,
// output: visible slope distribution D_s
double& ds_sx0, double& ds_sy0, double& ds_varx, double& ds_vary, double& ds_cxy) {
// pre-computations
double cos_theta = cos(theta);
double sin_theta = sin(theta);
double cot_theta = cos_theta / sin_theta;
double cos_phi = cos(phi);
double sin_phi = sin(phi);

// rotate p_s statistics to view-oriented basis
double ps_para0 = ps_sx0 * cos_phi + ps_sy0 * sin_phi;
double ps_ortho0= -ps_sx0 * sin_phi + ps_sy0 * cos_phi;
double ps_varpara = ps_varx*cos_phi*cos_phi + ps_vary*sin_phi*sin_phi + 2*ps_cxy*cos_phi*sin_phi;
double ps_varortho= ps_varx*sin_phi*sin_phi + ps_vary*cos_phi*cos_phi - 2*ps_cxy*cos_phi*sin_phi;
double ps_cparaortho= (ps_vary-ps_varx)*cos_phi*sin_phi + ps_cxy * (cos_phi*cos_phi-sin_phi*sin_phi);

// compute E1 and E2
double E1 = sqrt(0.5*ps_varpara/M_PI) * exp(-0.5 * (cot_theta-ps_para0)*(cot_theta-ps_para0)/ps_varpara);
double E2 = 1.0 - 0.5 * erfc( (cot_theta-ps_para0) / sqrt(2.0*ps_varpara) );

// compute Integrals I
double I = E2;
double I_para = -E1 + ps_para0 * E2;
double I_ortho = -ps_cparaortho/ps_varpara*E1 + ps_ortho0 * E2;
double I_para2 = -cot_theta * E1 + ps_varpara*E2 + ps_para0*I_para;
double I_ortho2 = -ps_cparaortho/ps_varpara*(ps_ortho0+ps_cparaortho/ps_varpara*(cot_theta-ps_para0))*E1
+ ps_varortho*E2 + ps_ortho0*I_ortho;
double I_para_ortho = -ps_cparaortho/ps_varpara*cot_theta*E1 + ps_cparaortho*E2 + ps_ortho0*I_para;
double I_para2_ortho = -ps_cparaortho/ps_varpara*cot_theta*cot_theta*E1 + 2.0*ps_cparaortho*I_para
+ ps_ortho0*I_para2;
double I_para_ortho2 = -(ps_varpara*ps_varortho-ps_cparaortho*ps_cparaortho)/ps_varpara
+ (ps_ortho0+ps_cparaortho/ps_varpara*(cot_theta-ps_para0))*(ps_ortho0+
ps_cparaortho/ps_varpara*(cot_theta-ps_para0))*E1 + ps_para0 * I_ortho2 + 2.0 *
ps_cparaortho * I_ortho;
double I_para3 = 2.0 * ps_varpara*I_para - cot_theta*cot_theta * E1 + ps_para0*I_para2;

// compute D_s in view-oriented basis
double N = -sin_theta*I_para + cos_theta*I;
double ds_para0 = (-sin_theta*I_para2 + cos_theta*I_para) / N;
double ds_ortho0 = (-sin_theta*I_para_ortho + cos_theta*I_ortho) / N;
double ds_varpara = (-sin_theta*I_para3 + cos_theta*I_para2) / N - ds_para0*ds_para0;
double ds_varortho = (-sin_theta*I_para_ortho2 + cos_theta*I_ortho2) / N - ds_ortho0*ds_ortho0;
double ds_cparaortho = (-sin_theta*I_para2_ortho + cos_theta*I_para_ortho) / N - ds_para0*ds_ortho0;

// rotate D_s statistics to canonic basis
ds_sx0 = ds_para0 * cos_phi - ds_ortho0 * sin_phi;

```



```
ds_sy0 = ds_para0 * sin_phi + ds_ortho0 * cos_phi;  
ds_varx = ds_varpara * cos_phi*cos_phi + ds_varortho * sin_phi*sin_phi - 2 * ds_cparaortho*cos_phi*sin_phi;  
ds_vary = ds_varpara * sin_phi*sin_phi + ds_varortho * cos_phi*cos_phi + 2 * ds_cparaortho*cos_phi*sin_phi;  
ds_cxy = (ds_varpara-ds_varortho)*cos_phi*sin_phi + ds_cparaortho * (cos_phi*cos_phi - sin_phi*sin_phi);  
}
```

Annexe F

Résultats supplémentaires pour l'importance sampling (chapitre 11)

F.1 Generic Sampling Algorithm

Generic Algorithm. The idea is that if P^{22} is shape-invariant, and if there is a way to simulate $(x_{\tilde{m}}, y_{\tilde{m}})$ with the PDF $P_{\omega_i}^{22}$ with roughness $\alpha = 1$, then it is possible to use it to simulate the same distribution with any parameters α_x and α_y . We recall Algorithm 7 from chapter 11 :

Algorithm 9 Simulate $D_{\omega_i}(\omega_m)$ (Smith profile)

$\omega'_i \leftarrow \mathcal{S}^{\alpha_x, \alpha_y}(\omega_i)$	▷ 1. stretch
$x_{\tilde{m}} \leftarrow C_{\omega'_i}^{2-1}(\mathcal{U}_1, 1, 1)$	▷ 2.a. simulate $P_{\omega'_i}^{2-}(x_{\tilde{m}}, 1, 1)$
$y_{\tilde{m}} \leftarrow C^{2 2-1}(\mathcal{U}_2 x_{\tilde{m}}, 1, 1)$	▷ 2.b. simulate $P^{2 2}(y_{\tilde{m}} x_{\tilde{m}}, 1, 1)$
$\begin{pmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{pmatrix} \leftarrow \begin{pmatrix} \cos \phi'_i & -\sin \phi'_i \\ \sin \phi'_i & \cos \phi'_i \end{pmatrix} \begin{pmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{pmatrix}$	▷ 3. rotate
$\begin{pmatrix} x_{\tilde{m}} \\ y_{\tilde{m}} \end{pmatrix} \leftarrow \begin{pmatrix} \alpha_x x_{\tilde{m}} \\ \alpha_y y_{\tilde{m}} \end{pmatrix}$	▷ 4. unstretch
$\omega_m \leftarrow \frac{(-x_{\tilde{m}}, -y_{\tilde{m}}, 1)}{\sqrt{x_{\tilde{m}}^2 + y_{\tilde{m}}^2 + 1}}$	▷ 5. compute normal

Objectives. In this document, we show how to compute the inverse CDFs

$$x_{\tilde{m}} = C_{\omega'_i}^{2-1}(\mathcal{U}_1, 1, 1),$$

$$y_{\tilde{m}} = C^{2|2-1}(\mathcal{U}_2|x_{\tilde{m}}, 1, 1),$$

analytically with Beckmann (Algorithm 10 of this document) and GGX (Algorithm 11 of this document) distributions.

F.1.1 C++ Implementation

We describe a C++ implementation of the paper’s Algorithm 4 and of the associated precomputations steps. We implement a virtual class **VNDFSampler** (Visible Normal Distribution Function Sampler). The API contains one public method **sample()**, which takes as arguments the incident direction ω_i , anisotropic roughness α_x and α_y , and two uniform random numbers \mathcal{U}_1 and \mathcal{U}_2 . It generates a microfacet normal ω_m with PDF $D_{\omega_i}(\omega_m, \alpha_x, \alpha_y)$.

A child class must implement the protected virtual method **sample11()**, which returns a slope vector $(x_{\bar{m}}, y_{\bar{m}})$ sampled from the stretched configuration with roughness $\alpha = 1$, i.e. from distribution $P_{\omega_i}^{22}(x_{\bar{m}}, y_{\bar{m}}, 1, 1)$. The implementation of the child classes for Beckmann and GGX distributions are provided in Sections F.3 and F.4.

```
class VNDFSampler
{
public:
    // generate a view dependent normal
    void sample(
        // input
        const double omega_i[3], // incident direction
        const double alpha_x, const double alpha_y, // anisotropic roughness
        const double U1, const double U2, // random numbers
        // output
        double omega_m[3] // normal
    );

protected:
    // method implemented by child class
    // inverse slope CDF
    virtual void sample11(
        // input
        const double theta_i, // incident direction
        double U1, double U2, // random numbers
        // output
        double& slope_x, double& slope_y // slopes
    ) = 0;
};
```

```

void VNDFSampler::sample(
    // input
    const double omega_i[3], // incident direction
    const double alpha_x, const double alpha_y, // anisotropic roughness
    const double U1, const double U2, // random numbers
    // output
    double omega_m[3]) // micronormal
{
    // 1. stretch omega_i
    double omega_i_[3];
    omega_i_[0] = alpha_x * omega_i[0];
    omega_i_[1] = alpha_y * omega_i[1];
    omega_i_[2] = omega_i[2];
    // normalize
    double inv_omega_i = 1.0 / sqrt(omega_i_[0]*omega_i_[0] + omega_i_[1]*omega_i_[1] + omega_i_[2]*omega_i_[2]);
    omega_i_[0] *= inv_omega_i;
    omega_i_[1] *= inv_omega_i;
    omega_i_[2] *= inv_omega_i;
    // get polar coordinates of omega_i_
    double theta_ = 0.0;
    double phi_ = 0.0;
    if (omega_i_[2] < 0.99999)
    {
        theta_ = acos(omega_i_[2]);
        phi_ = atan2(omega_i_[1], omega_i_[0]);
    }

    // 2. simulate P22_{omega_i}(x_slope, y_slope, 1, 1)
    double slope_x, slope_y;
    sample11( theta_,
        U1, U2,
        slope_x, slope_y);

    // 3. rotate
    double tmp = cos(phi_)*slope_x - sin(phi_)*slope_y;
    slope_y = sin(phi_)*slope_x + cos(phi_)*slope_y;
    slope_x = tmp;

    // 4. unstretch
    slope_x = alpha_x * slope_x;
    slope_y = alpha_y * slope_y;

    // 5. compute normal
    double inv_omega_m = sqrt(slope_x*slope_x + slope_y*slope_y + 1.0);
    omega_m[0] = -slope_x/inv_omega_m;
    omega_m[1] = -slope_y/inv_omega_m;
    omega_m[2] = 1.0/inv_omega_m;
}

```

F.2 Sparing Random Numbers

We use several conditional branchings in our sampling algorithms. But we can spare the random numbers used for these branchings. Indeed, each time a random number \mathcal{U} is used in a branching with probability p , its value can be used again by renormalizing it :

```
if  $\mathcal{U} < p$  then
   $\mathcal{U} \leftarrow \frac{\mathcal{U}}{p}$                                 ▷ renormalize  $\mathcal{U}$ 
  do something with  $\mathcal{U}$ 
else
   $\mathcal{U} \leftarrow \frac{\mathcal{U}-p}{1-p}$                     ▷ renormalize  $\mathcal{U}$ 
  do something else with  $\mathcal{U}$ 
end if
```

F.3 Sampling Beckmann Distributions

F.3.1 The Beckmann Distribution of Slopes

The Beckmann distribution of slopes with $\alpha = 1$ is the 2D Gaussian

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) = \frac{1}{\pi} \exp(-x_{\tilde{m}}^2 - y_{\tilde{m}}^2).$$

F.3.2 Simulate $P_{\omega_i}^{2-}(x_{\tilde{m}})$

The Marginalized Distribution of Slopes is the 1D Gaussian distribution

$$\begin{aligned} P^{2-}(x_{\tilde{m}}) &= \int_{-\infty}^{+\infty} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dy_{\tilde{m}} \\ &= \frac{1}{\sqrt{\pi}} \exp(-x_{\tilde{m}}^2), \end{aligned}$$

and its CDF is :

$$\begin{aligned} C^{2-}(x_{\tilde{m}}) &= \int_{-\infty}^{x_{\tilde{m}}} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\ &= \frac{1}{2} + \frac{1}{2} \operatorname{erf}(x_{\tilde{m}}). \end{aligned}$$

The Marginalized Distribution of Visible Slopes is

$$\begin{aligned} P_{\omega_i}^{2-}(x_{\tilde{m}}) &= \frac{(-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\tilde{m}})}{\int_{-\infty}^{+\infty} (-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}}} \\ &= \frac{G_1(\omega_i)}{\cos \theta_i} (-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\tilde{m}}), \end{aligned}$$

and the associated CDF is

$$\begin{aligned} C_{\omega_i}^{2-}(x_{\tilde{m}}) &= \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\infty}^{x_{\tilde{m}}} (-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}} \\ &= \frac{G_1(\omega_i) \tan \theta_o}{2\sqrt{\pi}} \exp(-x_{\tilde{m}}^2) + G_1(\omega_i) \left(\frac{1}{2} + \frac{1}{2} \operatorname{erf}(x_{\tilde{m}}) \right). \end{aligned}$$

The problem is that this CDF is not directly invertible.

Choosing an interval. The CDF $C_{\omega_i}^{2-}(x_{\tilde{m}})$ is invertible in the interval $] -\infty, -\cot \theta_i]$ and we use the anti-symmetry of the projection factor to invert it in $] -\cot \theta_i, \cot \theta_i]$. We choose randomly the interval $] -\infty, -\cot \theta_i]$ or $] -\cot \theta_i, \cot \theta_i]$ according to their weights given by the CDF value at $-\cot \theta_i$:

```

if  $\mathcal{U} \leq C_{\omega_i}^{2-}(-\cot \theta_i)$  then
  simulate  $x_{\tilde{m}} \in ] -\infty, -\cot \theta_i]$ 
else
  simulate  $x_{\tilde{m}} \in ] -\cot \theta_i, \cot \theta_i]$ 
end if

```

and we have :

$$\begin{aligned}
& C_{\omega_i}^{2-}(-\cot \theta_i) \\
&= \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\infty}^{-\cot \theta_i} (-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\
&= \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\infty}^{\cot \theta_i} (-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\
&\quad - \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\cot \theta_i}^{\cot \theta_i} (-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\
&= 1 - \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\cot \theta_i}^{\cot \theta_i} (-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\
&= 1 - G_1(\omega_i) \int_{-\cot \theta_i}^{\cot \theta_i} P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} \\
&= 1 - G_1(\omega_i) (C^{2-}(\cot \theta_i) - C^{2-}(-\cot \theta_i)) \\
&= 1 - G_1(\omega_i) \operatorname{erf}(\cot \theta_i).
\end{aligned}$$

Simulate $P_{\omega_i}^{2-}(x_{\tilde{m}})$ in $] -\infty, -\cot \theta_i]$.

If $x_{\tilde{m}} \in] -\infty, -\cot \theta_i]$, we have $-x_{\tilde{m}} \sin \theta_i + \cos \theta_i > 0$, so we can drop the heaviside term and separate the integral :

$$\begin{aligned} C_{\omega_i}^{2-}(x_{\tilde{m}}) &= \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\infty}^{x_{\tilde{m}}} (-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}} \\ &= \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\infty}^{x_{\tilde{m}}} (-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}} \\ &= \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\infty}^{x_{\tilde{m}}} (-x'_{\tilde{m}} \sin \theta_i) P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}} + \frac{G_1(\omega_i)}{\cos \theta_i} \int_{-\infty}^{x_{\tilde{m}}} \cos \theta_i P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}} \\ &= \frac{G_1(\omega_i)}{\cos \theta_i} (f_1(x) + f_2(x)), \end{aligned}$$

where

$$\begin{aligned} f_1(x_{\tilde{m}}) &= \int_{-\infty}^{x_{\tilde{m}}} (-x'_{\tilde{m}} \sin \theta_i) P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}}, \\ f_2(x_{\tilde{m}}) &= \int_{-\infty}^{x_{\tilde{m}}} \cos \theta_i P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}}. \end{aligned}$$

We can sample with PDF $P_{\omega_i}^{2-}(x_{\tilde{m}})$ by choosing either f_1 or f_2 according to their respective weight ratios, and inverting it. Their weights are given by :

$$\begin{aligned} w_1 = f_1(-\cot \theta_i) &= \int_{-\infty}^{-\cot \theta_i} (-x_{\tilde{m}} \sin \theta_i) P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} = \frac{\sin \theta_i}{2\sqrt{\pi}} \exp(-\cot^2 \theta_i), \\ w_2 = f_2(-\cot \theta_i) &= \int_{-\infty}^{-\cot \theta_i} \cos \theta_i P^{2-}(x_{\tilde{m}}) dx_{\tilde{m}} = \cos \theta_i \left(\frac{1}{2} - \frac{1}{2} \operatorname{erf}(\cot \theta_i) \right). \end{aligned}$$

The branching probability p is given by :

$$p = \frac{w_1}{w_1 + w_2},$$

and the functions to inverse are f_1 and f_2 divided by their respective weights to transform them into PDFs before the inversion :

$$\begin{aligned} \left(\frac{1}{w_1} f_1 \right)^{-1}(\mathcal{U}) &= -\sqrt{-\log(\mathcal{U} \exp(-\cot^2 \theta_i))} \\ \left(\frac{1}{w_2} f_2 \right)^{-1}(\mathcal{U}) &= \operatorname{erf}^{-1}(\mathcal{U} - 1 - \mathcal{U} \operatorname{erf}(\cot \theta_i)). \end{aligned}$$

The algorithm we use to simulate $P_{\omega_i}^{2-}(x_{\tilde{m}})$ in $] -\infty, -\cot \theta_i]$ is thus :

```

 $w_1 \leftarrow \frac{\sin \theta_i}{2\sqrt{\pi}} \exp(-\cot^2 \theta_i)$ 
 $w_2 \leftarrow \cos \theta_i \left(\frac{1}{2} - \frac{1}{2} \operatorname{erf}(\cot \theta_i)\right)$ 
 $p = \frac{w_1}{w_1 + w_2}$ 
if  $\mathcal{U} \leq p$  then
     $\mathcal{U} \leftarrow \frac{\mathcal{U}}{p}$  ▷ renormalize  $\mathcal{U}$ 
     $x_{\bar{m}} \leftarrow -\sqrt{-\log(\mathcal{U} \exp(-\cot^2 \theta_i))}$ 
else
     $\mathcal{U} \leftarrow \frac{\mathcal{U} - p}{1 - p}$  ▷ renormalize  $\mathcal{U}$ 
     $x_{\bar{m}} \leftarrow \operatorname{erf}^{-1}(\mathcal{U} - 1 - \mathcal{U} \operatorname{erf}(\cot \theta_i))$ 
end if

```

Simulate $P_{\omega_i}^{2-}(x_{\bar{m}})$ **in** $] -\cot \theta_i, \cot \theta_i]$.

The PDF of a *couple* of slope values $(-x_{\bar{m}}, x_{\bar{m}})$ in the interval $] -\cot \theta_i, \cot \theta_i]$ is :

$$(x_{\bar{m}} \sin \theta_i + \cos \theta_i) P^{2-}(-x_{\bar{m}}) + (-x_{\bar{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\bar{m}}) = 2 \cos \theta_i P^{2-}(x_{\bar{m}}).$$

The idea is similar to the sampling scheme for the V-cavity profile. We start by generate a couple $(-x_{\bar{m}}, x_{\bar{m}})$ with PDF $P^{2-}(x_{\bar{m}})$ in the interval $] -\cot \theta_i, \cot \theta_i]$:

$$\begin{aligned} x_{\bar{m}} &= C^{2-}{}^{-1} \left((1 - \mathcal{U}) C^{2-}(-\cot \theta_i) + \mathcal{U} C^{2-}(\cot \theta_i) \right) \\ &= \operatorname{erf}^{-1} \left((-1 + 2\mathcal{U}) \operatorname{erf}(\cot \theta_i) \right), \end{aligned}$$

and then we choose between $-x_{\bar{m}}$ and $x_{\bar{m}}$ with a probability proportionnal to their projected areas $(x_{\bar{m}} \sin \theta_i + \cos \theta_i)$ and $(-x_{\bar{m}} \sin \theta_i + \cos \theta_i)$. In practice, we swap $x_{\bar{m}}$ and $-x_{\bar{m}}$ with probability :

$$\begin{aligned} p &= \frac{-x_{\bar{m}} \sin \theta_i + \cos \theta_i}{-x_{\bar{m}} \sin \theta_i + \cos \theta_i + x_{\bar{m}} \sin \theta_i + \cos \theta_i} \\ &= \frac{-x_{\bar{m}} \sin \theta_i + \cos \theta_i}{2 \cos \theta_i}. \end{aligned}$$

The algorithm we use to simulate $P_{\omega_i}^{2-}(x_{\bar{m}})$ in $] -\cot \theta_i, \cot \theta_i]$ is thus :

```

 $x_{\bar{m}} \leftarrow \operatorname{erf}^{-1} \left( (-1 + 2\mathcal{U}_1) \operatorname{erf}(\cot \theta_i) \right)$ 
if  $\mathcal{U}_2 > \frac{-x_{\bar{m}} \sin \theta_i + \cos \theta_i}{2 \cos \theta_i}$  then
     $x_{\bar{m}} \leftarrow -x_{\bar{m}}$ 
end if

```

Note that we need 2 random numbers \mathcal{U}_1 and \mathcal{U}_2 in this algorithm. In the final algorithm, we renormalize \mathcal{U}_2 which is used here for branching only and use it again in the simulation of $P^{2|2}(y_{\bar{m}}|x_{\bar{m}})$.

F.3.3 Simulate $P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}})$

The conditional PDF of slopes is the 1D Gaussian distribution

$$\begin{aligned} P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}) &= \frac{P^{22}(x_{\tilde{m}}, y_{\tilde{m}})}{\int_{-\infty}^{+\infty} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dy_{\tilde{m}}} \\ &= \frac{1}{\sqrt{\pi}} \exp(-y_{\tilde{m}}^2), \end{aligned}$$

and the associated CDF is

$$\begin{aligned} C^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}) &= \int_{-\infty}^{y_{\tilde{m}}} P^{2-}(y'_{\tilde{m}}|x_{\tilde{m}}) dy'_{\tilde{m}} \\ &= \frac{1}{2} + \frac{1}{2} \operatorname{erf}(y_{\tilde{m}}), \end{aligned}$$

which can be inverted :

$$\begin{aligned} \mathcal{U} &= C^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}) \\ \Rightarrow y_{\tilde{m}} &= \operatorname{erf}^{-1}(2\mathcal{U} - 1). \end{aligned}$$

F.3.4 Simulate $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$ for Beckmann Distributions

The complete algorithm is

Algorithm 10 Simulate $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$ for Beckmann distributions

$$C \leftarrow 1 - G_1(\omega_i) \operatorname{erf}(\cot \theta_i)$$

if $\mathcal{U}_1 \leq C$ **then**

$$\mathcal{U}_1 \leftarrow \frac{\mathcal{U}}{C}$$

▷ simulate $P_{\omega_i}^{22}(x_{\tilde{m}})$ in $] -\infty, -\cot \theta_i]$

▷ renormalize \mathcal{U}_1

$$w_1 \leftarrow \frac{\sin \theta_i}{2\sqrt{\pi}} \exp(-\cot^2 \theta_i)$$

$$w_2 \leftarrow \cos \theta_i \left(\frac{1}{2} - \frac{1}{2} \operatorname{erf}(\cot \theta_i) \right)$$

$$p = \frac{w_1}{w_1 + w_2}$$

if $\mathcal{U}_1 \leq p$ **then**

$$\mathcal{U}_1 \leftarrow \frac{\mathcal{U}}{p}$$

▷ renormalize \mathcal{U}_1

$$x_{\tilde{m}} \leftarrow -\sqrt{-\log(\mathcal{U}_1 \exp(-\cot^2 \theta_i))}$$

else

$$\mathcal{U}_1 \leftarrow \frac{\mathcal{U} - p}{1 - p}$$

▷ renormalize \mathcal{U}_1

$$x_{\tilde{m}} \leftarrow \operatorname{erf}^{-1}(\mathcal{U}_1 - 1 - \mathcal{U}_1 \operatorname{erf}(\cot \theta_i))$$

end if

else

▷ simulate $P_{\omega_i}^{22}(x_{\tilde{m}})$ in $] -\cot \theta_i, \cot \theta_i]$

$$\mathcal{U}_1 \leftarrow \frac{\mathcal{U}_1 - C}{1 - C}$$

▷ renormalize \mathcal{U}_1

$$x_{\tilde{m}} \leftarrow \operatorname{erf}^{-1}((-1 + 2\mathcal{U}_1) \operatorname{erf}(\cot \theta_i))$$

if $\mathcal{U}_2 > \frac{-x_{\tilde{m}} \sin \theta_i + \cos \theta_i}{2 \cos \theta_i}$ **then**

$$\mathcal{U}_2 \leftarrow \frac{\mathcal{U}_2 - p}{1 - p}$$

▷ renormalize \mathcal{U}_2

$$x_{\tilde{m}} \leftarrow -x_{\tilde{m}}$$

else

$$\mathcal{U}_2 \leftarrow \frac{\mathcal{U}_2}{p}$$

▷ renormalize \mathcal{U}_2

end if

end if

$$y_{\tilde{m}} \leftarrow \operatorname{erf}^{-1}(2\mathcal{U}_2 - 1)$$

▷ simulate $P^{22}(y_{\tilde{m}} | x_{\tilde{m}})$

F.3.5 C++ Implementation

The C++ implementation of Algorithm 10 is

```
class VNDFSamplerBeckmann : public VNDFSampler
{
protected:
    virtual void sample11(
        // input
        const double theta_i, // incident direction
        double U1, double U2, // random numbers
        // output
        double& slope_x, double& slope_y // slopes
    )
    {
        // special case (normal incidence)
        if(theta_i < 0.0001)
        {
            const double r = sqrt(-log(U1));
            const double phi = 6.28318530718 * U2;
            slope_x = r * cos(phi);
            slope_y = r * sin(phi);
            return;
        }

        // precomputations
        const double sin_theta_i = sin(theta_i);
        const double cos_theta_i = cos(theta_i);
        const double tan_theta_i = sin_theta_i/cos_theta_i;
        const double a = 1.0 / tan_theta_i;
        const double erf_a = erf(a);
        const double exp_a2 = exp(-a*a);
        const double SQRT_PI_INV = 0.56418958354;
        const double Lambda = 0.5*(erf_a-1) + 0.5*SQRT_PI_INV*exp_a2/a;
        const double G1 = 1.0 / (1.0 + Lambda); // masking
        const double C= 1.0 - G1 * erf_a;

        // simulate slope X
        if(U1 < C)
        {
            // rescale U1
            U1 = U1 / C;

            const double w_1 = 0.5 * SQRT_PI_INV * sin_theta_i * exp_a2;
            const double w_2 = cos_theta_i * (0.5 - 0.5*erf_a);
            const double p = w_1 / (w_1+w_2);

            if(U1 < p)
            {
                U1 = U1 / p;
                slope_x = -sqrt(-log(U1*exp_a2));
            }
            else
            {
                U1 = (U1-p) / (1.0-p);
                slope_x = erfinv(U1-1.0-U1*erf_a);
            }
        }
    }
};
```

```

    }
}
else
{
    // rescale U1
    U1 = (U1-C) / (1.0-C);

    slope_x = erfinv((-1.0+2.0*U1)*erf_a);

    const double p = (-slope_x*sin_theta_i + cos_theta_i) / (2.0*cos_theta_i);
    if (U2 > p)
    {
        slope_x = -slope_x;
        U2 = (U2-p) / (1.0-p);
    }
    else
        U2 = U2 / p;
}

// simulate slope Y
slope_y = erfinv(2.0*U2-1.0);
}
};

```

F.3.6 Results

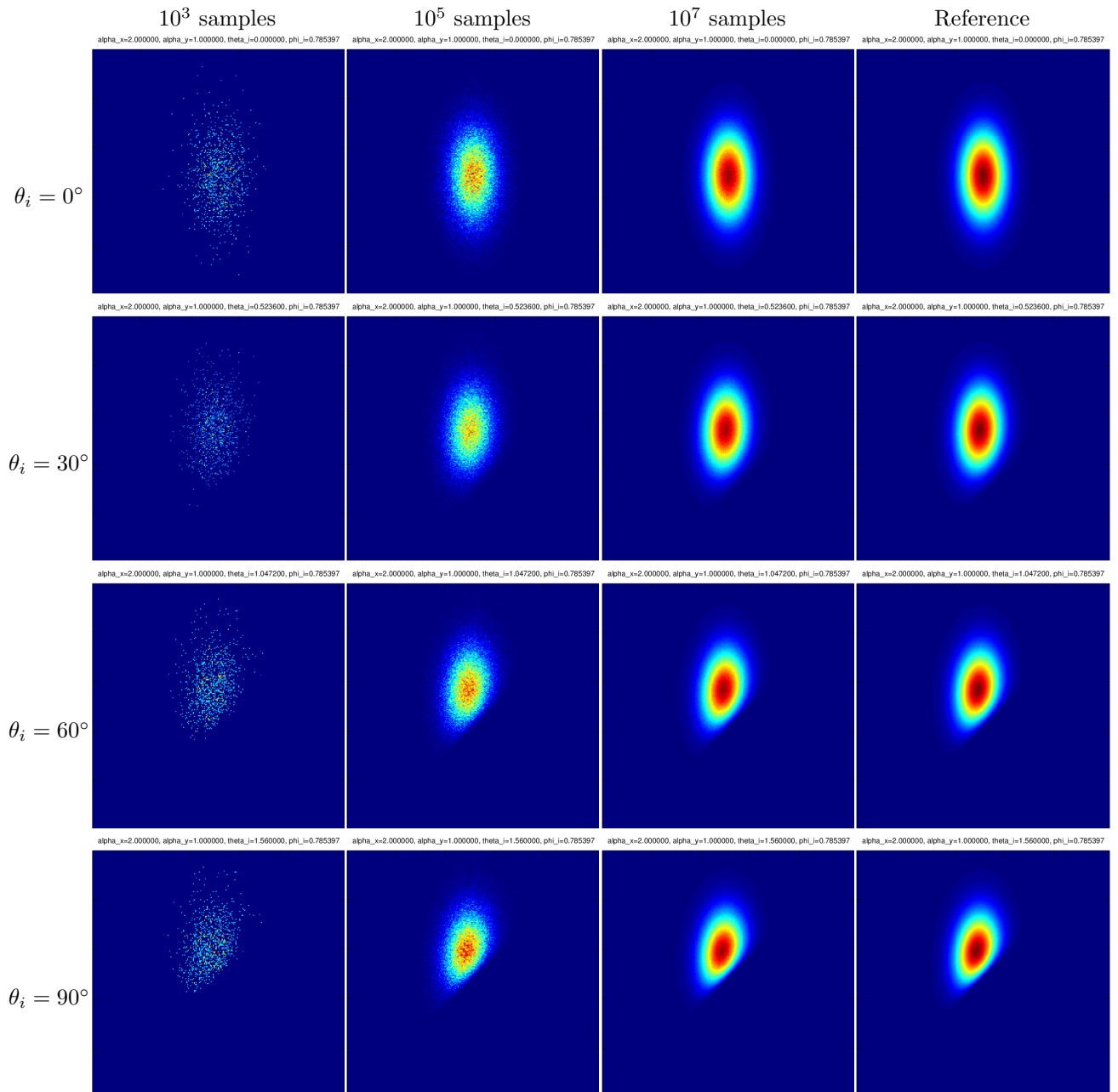


FIGURE F.1 – Sampling the distribution of visible slopes $P_{\omega_i}^{22}$ using our C++ implementation, where P^{22} is a Beckmann distribution with anisotropic roughness $\alpha_x = 2$ and $\alpha_y = 1$. The azimuthal angle of incidence is $\phi_i = 45^\circ$.

F.4 Sampling GGX Distributions

F.4.1 The GGX Distribution of Slopes

The GGX distributions of slopes with $\alpha = 1$ is

$$P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) = \frac{1}{\pi} \frac{1}{(1 + x_{\tilde{m}}^2 + y_{\tilde{m}}^2)^2}.$$

F.4.2 Simulate $P_{\omega_i}^{2-}(x_{\tilde{m}})$

The Marginalized Distribution of Slopes is

$$\begin{aligned} P^{2-}(x_{\tilde{m}}) &= \int_{-\infty}^{\infty} P^{22}(x_{\tilde{m}}, y_{\tilde{m}}) dy_{\tilde{m}} \\ &= \int_{-\infty}^{\infty} \frac{1}{\pi} \frac{1}{(1 + x_{\tilde{m}}^2 + y_{\tilde{m}}^2)^2} dy_{\tilde{m}} \\ &= \frac{1}{2(x_{\tilde{m}}^2 + 1)^{3/2}}. \end{aligned}$$

The Marginalized Distribution of Visible Slopes is

$$\begin{aligned} P_{\omega_i}^{2-}(x_{\tilde{m}}) &= \frac{(-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\tilde{m}})}{\int_{-\infty}^{+\infty} (-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x'_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}}} \\ &= \frac{G_1(\omega_i)}{\cos \theta_i} (-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) P^{2-}(x_{\tilde{m}}) \\ &= \frac{G_1(\omega_i)}{\cos \theta_i} (-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) \chi^+(-x_{\tilde{m}} \sin \theta_i + \cos \theta_i) \frac{1}{2(x_{\tilde{m}}^2 + 1)^{3/2}}, \end{aligned}$$

and the associated CDF is :

$$\begin{aligned} C_{\omega_i}^{2-}(x_{\tilde{m}}) &= \int_{-\infty}^{x_{\tilde{m}}} P_{\omega_i}^{2-}(x'_{\tilde{m}}) dx'_{\tilde{m}} \\ &= \frac{G_1(\omega_i)}{2 \cos \theta_i} \left(\frac{\sin \theta_i + x_{\tilde{m}} \cos \theta_i}{\sqrt{1 + x_{\tilde{m}}^2}} + \cos \theta_i \right). \end{aligned}$$

Inversion of the CDF. We solve

$$\begin{aligned} \mathcal{U} &= C_{\omega_i}^{2-}(x_{\tilde{m}}) = \frac{G_1(\omega_i)}{2 \cos \theta_i} \left(\frac{\sin \theta_i + x_{\tilde{m}} \cos \theta_i}{\sqrt{1 + x_{\tilde{m}}^2}} + \cos \theta_i \right) \\ \Rightarrow \frac{2\mathcal{U}}{G_1} - 1 &= \frac{\tan \theta_i + x_{\tilde{m}}}{\sqrt{1 + x_{\tilde{m}}^2}} \\ \Rightarrow A &= \frac{B + x_{\tilde{m}}}{\sqrt{1 + x_{\tilde{m}}^2}}, \end{aligned}$$

with $A = \frac{2\mathcal{U}}{G_1} - 1$ and $B = \tan \theta_i$. The solution is

$$x_{\tilde{m}} = \begin{cases} x_{\tilde{m}1} & \text{if } A < 0 \text{ or } x_{\tilde{m}2} > \cot \theta_i \\ x_{\tilde{m}2} & \text{else,} \end{cases}$$

where

$$x_{\tilde{m}1} = \frac{B}{A^2 - 1} - \sqrt{\frac{B^2}{(A^2 - 1)^2} - \frac{A^2 - B^2}{A^2 - 1}},$$

$$x_{\tilde{m}2} = \frac{B}{A^2 - 1} + \sqrt{\frac{B^2}{(A^2 - 1)^2} - \frac{A^2 - B^2}{A^2 - 1}}.$$

The algorithm we use to sample $P_{\omega_i}^{2-}(x_{\tilde{m}})$ is thus

```

A ←  $\frac{2\mathcal{U}}{G_1(\omega_i)} - 1$ 
B = tan  $\theta_i$ 
 $x_{\tilde{m}1} \leftarrow \frac{B}{A^2-1} - \sqrt{\frac{B^2}{(A^2-1)^2} - \frac{A^2-B^2}{A^2-1}}$ 
 $x_{\tilde{m}2} \leftarrow \frac{B}{A^2-1} + \sqrt{\frac{B^2}{(A^2-1)^2} - \frac{A^2-B^2}{A^2-1}}$ 
if  $A < 0$  or  $x_{\tilde{m}2} > \cot \theta_i$  then
     $x_{\tilde{m}} \leftarrow x_{\tilde{m}1}$ 
else
     $x_{\tilde{m}} \leftarrow x_{\tilde{m}2}$ 
end if

```

F.4.3 Simulate $P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}})$

The conditional PDF is given by :

$$P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}) = \frac{\frac{1}{(1+x_{\tilde{m}}^2+y_{\tilde{m}}^2)^2}}{\int_{-\infty}^{\infty} \frac{1}{(1+x_{\tilde{m}}^2+y'_{\tilde{m}})^2} dy'_{\tilde{m}}},$$

and the associated conditional CDF is :

$$C^{2|2}(y_{\tilde{m}}|x_{\tilde{m}}) = \frac{\int_{-\infty}^{y_{\tilde{m}}} \frac{1}{(1+x_{\tilde{m}}^2+y'_{\tilde{m}})^2} dy'_{\tilde{m}}}{\int_{-\infty}^{\infty} \frac{1}{(1+x_{\tilde{m}}^2+y'_{\tilde{m}})^2} dy'_{\tilde{m}}}.$$

Unfortunately, this CDF is not directly invertible.

Change of variable. With a change of variable we make the formulation independent of $x_{\tilde{m}}$:

$$\begin{aligned} C^{22}(y_{\tilde{m}}|x_{\tilde{m}}) &= \frac{\int_{-\infty}^{y_{\tilde{m}}} \frac{1}{(1+x_{\tilde{m}}^2+y'_{\tilde{m}})^2} dy'_{\tilde{m}}}{\int_{-\infty}^{\infty} \frac{1}{(1+x_{\tilde{m}}^2+y'_{\tilde{m}})^2} dy'_{\tilde{m}}} \\ &= \frac{\int_{-\infty}^{y_{\tilde{m}}} \frac{1}{(1+\frac{y'_{\tilde{m}}}{a})^2} dy'_{\tilde{m}}}{\int_{-\infty}^{\infty} \frac{1}{(1+\frac{y'_{\tilde{m}}}{a})^2} dy'_{\tilde{m}}} \end{aligned}$$

with $a = 1 + x_{\tilde{m}}^2$. Let be $z' = y'_{\tilde{m}}/\sqrt{a}$ and $z = y_{\tilde{m}}/\sqrt{a}$, and we get

$$\begin{aligned} C^{22}(y_{\tilde{m}}|x_{\tilde{m}}) &= C_z(z) \\ &= \frac{\int_{-\infty}^z \frac{1}{(1+z'^2)^2} dz'}{\int_{-\infty}^{\infty} \frac{1}{(1+z'^2)^2} dz'}. \end{aligned}$$

The sampling of z does not depend on $x_{\tilde{m}}$ anymore. We fit the inverse CDF C_z^{-1} of z and we solve

$$\begin{aligned} \mathcal{U} &= C^{22}(y_{\tilde{m}}|x_{\tilde{m}}) = C_z(z) \\ &\Rightarrow z = C_z^{-1}(\mathcal{U}) \\ &\Rightarrow y_{\tilde{m}} = C_z^{-1}(\mathcal{U}) \sqrt{1+x_{\tilde{m}}^2} \end{aligned}$$

We use a rational polynomial to fit C_z^{-1} :

$$C_z^{-1}(\mathcal{U}) = \frac{0.46341\mathcal{U} - 0.73369\mathcal{U}^2 + 0.27385\mathcal{U}^3}{0.597999 - \mathcal{U} + 0.309420\mathcal{U}^2 + 0.093073\mathcal{U}^3},$$

in the positive interval $[0, +\infty[$. To use this fit in the interval $] -\infty, +\infty[$, if $\mathcal{U} < \frac{1}{2}$, we compute $z = C_z^{-1}(2(\mathcal{U} - \frac{1}{2}))$, else if $\mathcal{U} \geq \frac{1}{2}$ we compute $z = C_z^{-1}(2(\frac{1}{2} - \mathcal{U}))$. In this way, we can sample $z \in] -\infty, \infty[$. This fit is plotted in Figure F.2.

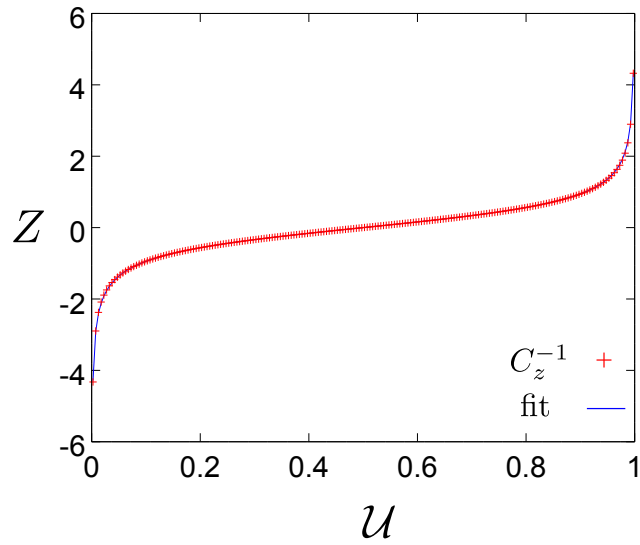


FIGURE F.2 – Plot of our fit of the inverse CDF C_z^{-1} .

The algorithm we use to simulate $P^{2|2}(y_{\tilde{m}}|x_{\tilde{m}})$ is thus

```

if  $U \leq \frac{1}{2}$  then
   $s = 1$ 
   $U \leftarrow 2(U - \frac{1}{2})$ 
else
   $s = -1$ 
   $U \leftarrow 2(\frac{1}{2} - U)$ 
end if
 $z \leftarrow s \frac{0.46341U - 0.73369U^2 + 0.27385U^3}{0.597999 - U + 0.309420U^2 + 0.093073U^3}$ 
 $y_{\tilde{m}} \leftarrow z \sqrt{1 + x_{\tilde{m}}^2}$ 

```

F.4.4 Simulate $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$ for GGX Distributions

Algorithm 11 Simulate $P_{\omega_i}^{22}(x_{\tilde{m}}, y_{\tilde{m}}, 1, 1)$ for GGX distributions

```

 $A \leftarrow \frac{2\mathcal{U}_1}{G_1(\omega_i)} - 1$ 
 $B = \tan \theta_i$ 
 $x_{\tilde{m}1} \leftarrow \frac{B}{A^2-1} - \sqrt{\frac{B^2}{(A^2-1)^2} - \frac{A^2-B^2}{A^2-1}}$ 
 $x_{\tilde{m}2} \leftarrow \frac{B}{A^2-1} + \sqrt{\frac{B^2}{(A^2-1)^2} - \frac{A^2-B^2}{A^2-1}}$ 
if  $A < 0$  or  $X_2 > \cot \theta_i$  then
     $x_{\tilde{m}} \leftarrow x_{\tilde{m}1}$ 
else
     $x_{\tilde{m}} \leftarrow x_{\tilde{m}2}$ 
end if

if  $\mathcal{U}_2 \leq \frac{1}{2}$  then
     $s = 1$ 
     $\mathcal{U}_2 \leftarrow 2(\mathcal{U}_2 - \frac{1}{2})$ 
else
     $s = -1$ 
     $\mathcal{U}_2 \leftarrow 2(\frac{1}{2} - \mathcal{U}_2)$ 
end if
 $z \leftarrow s \frac{0.46341\mathcal{U}_2 - 0.73369\mathcal{U}_2^2 + 0.27385\mathcal{U}_2^3}{0.597999 - \mathcal{U}_2 + 0.309420\mathcal{U}_2^2 + 0.093073\mathcal{U}_2^3}$ 
 $y_{\tilde{m}} \leftarrow z \sqrt{1 + x_{\tilde{m}}^2}$ 

```

F.4.5 C++ Implementation

The C++ implementation of Algorithm 11 is

```
class VNDFSamplerGGX : public VNDFSampler
{
protected:
    virtual void sample11(
        // input
        const double theta_i, // incident direction
        double U1, double U2, // random numbers
        // output
        double& slope_x, double& slope_y // slopes
    )
    {
        // special case (normal incidence)
        if(theta_i < 0.0001)
        {
            const double r = sqrt(U1/(1-U1));
            const double phi = 6.28318530718 * U2;
            slope_x = r * cos(phi);
            slope_y = r * sin(phi);
            return;
        }

        // precomputations
        const double tan_theta_i = tan(theta_i);
        const double a = 1 / (tan_theta_i);
        const double G1 = 2 / (1 + sqrt(1.0+1.0/(a*a)));

        // sample slope_x
        const double A = 2.0*U1/G1 - 1.0;
        const double tmp = 1.0 / (A*A-1.0);
        const double B = tan_theta_i;
        const double D = sqrt(B*B*tmp*tmp - (A*A-B*B)*tmp);
        const double slope_x_1 = B*tmp - D;
        const double slope_x_2 = B*tmp + D;
        slope_x = (A < 0 || slope_x_2 > 1.0/tan_theta_i) ? slope_x_1 : slope_x_2;

        // sample slope_y
        double S;
        if(U2 > 0.5)
        {
            S = 1.0;
            U2 = 2.0*(U2-0.5);
        }
        else
        {
            S = -1.0;
            U2 = 2.0*(0.5-U2);
        }
        const double z = (U2*(U2*(U2*0.27385-0.73369)+0.46341)) / (U2*(U2*(U2*0.093073+0.309420)-1.000000)+0.597999);
        slope_y = S * z * sqrt(1.0+slope_x*slope_x);
    }
};
```

F.4.6 Results

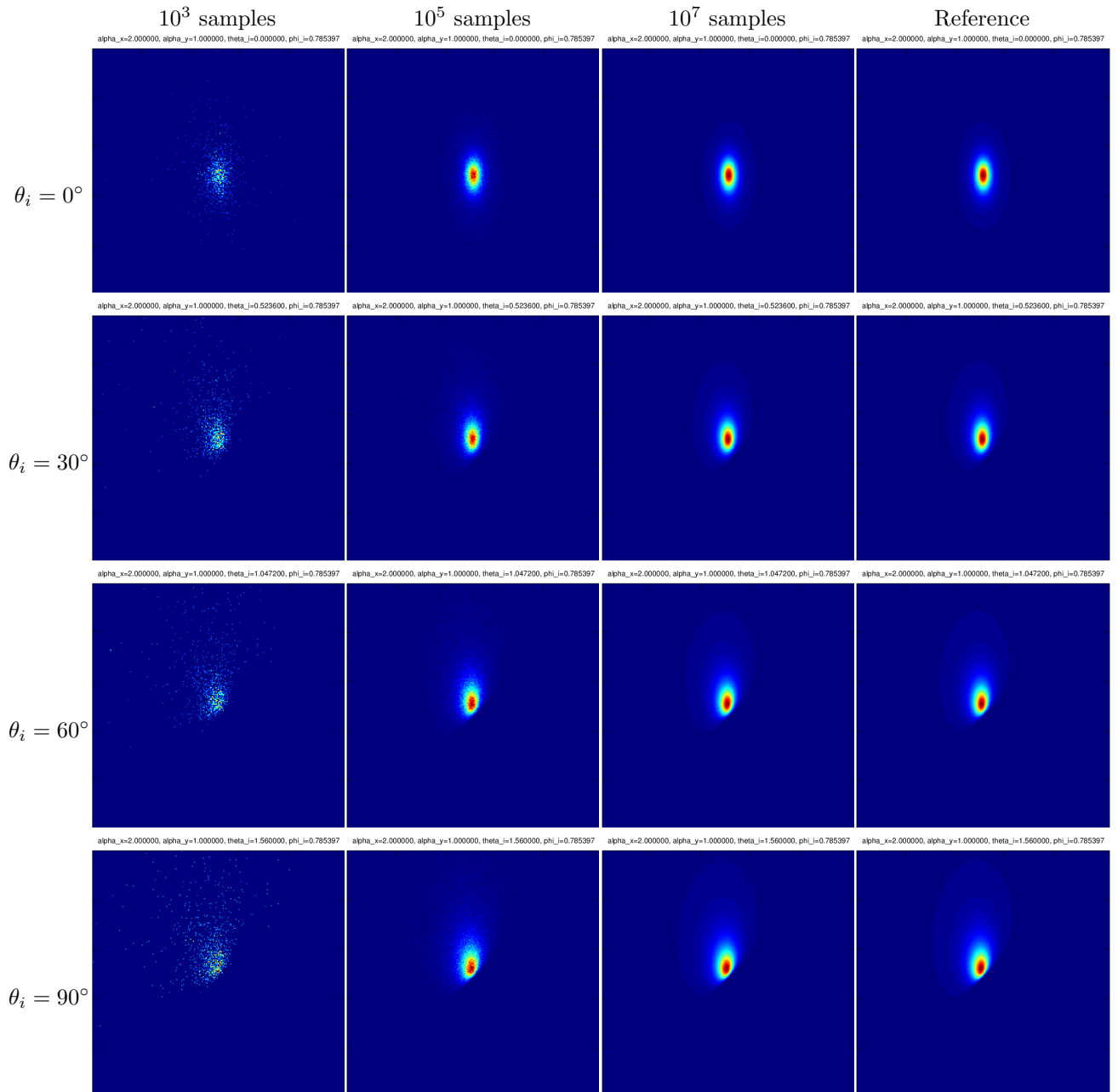


FIGURE F.3 – Sampling the distribution of visible slopes $P_{\omega_i}^{22}$ using our C++ implementation, where P^{22} is a GGX distribution with anisotropic roughness $\alpha_x = 2$ and $\alpha_y = 1$. The azimuthal angle of incidence is $\phi_i = 45^\circ$.

F.5 Convergence Comparisons

In this section we investigate in detail the convergence of the importance sampling scheme. To avoid perturbation due to the environment sampling, we use a perfectly white environment with $L(\omega_o) = 1$ (this is known as the White Furnace Test). We use importance sampling to solve the equation :

$$I = \int_{\Omega} f(\omega_i, \omega_o) |\omega_o \cdot \omega_g| d\omega_o,$$

whose solution is the average value of the shadowing term (≤ 1).

Table F.4 and F.5 compares the previous importance sampling technique with ours. Each plot shows 8 instances of the convergence process for this integral for both methods. We observe several properties :

- At normal incidence ($\theta_i = 0$) the techniques have equivalent convergence. In this case $\omega_i = \omega_g$ and thus $D_{\omega_i}(\omega_m) = (\omega_m \cdot \omega_g)D(\omega_m)$, and the techniques are equivalent.
- Our importance sampling is more efficient with incident angles $\theta_i > 1.0$ because this is where the effect of view dependence is important.
- These results shows also that the more the roughness α is important, the better our technique is. This is because view-dependent effects are the most important on bent micromorphals, whose presence depends on the roughness.

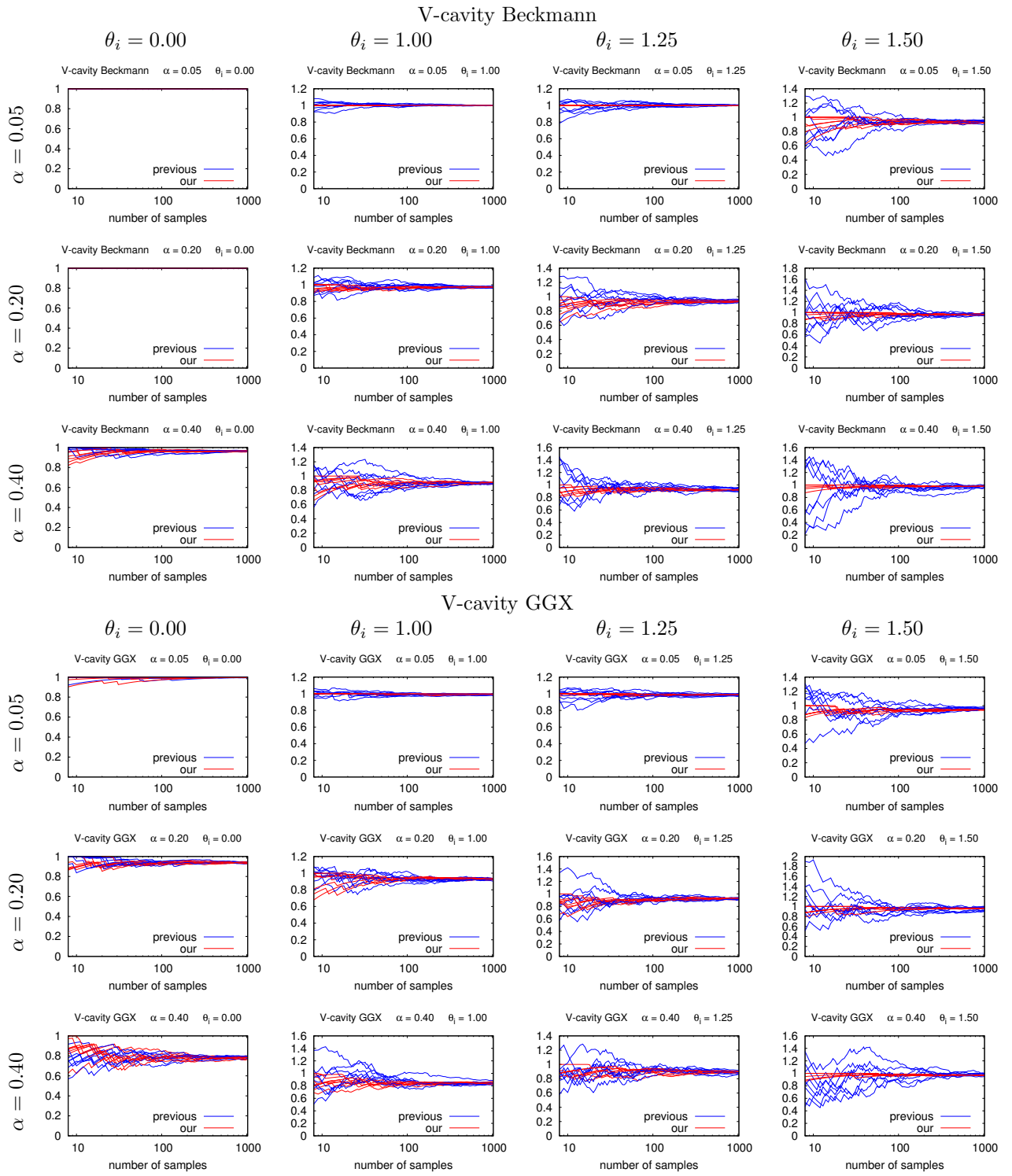


FIGURE F.4 – Importance sampling with the V-cavity model.

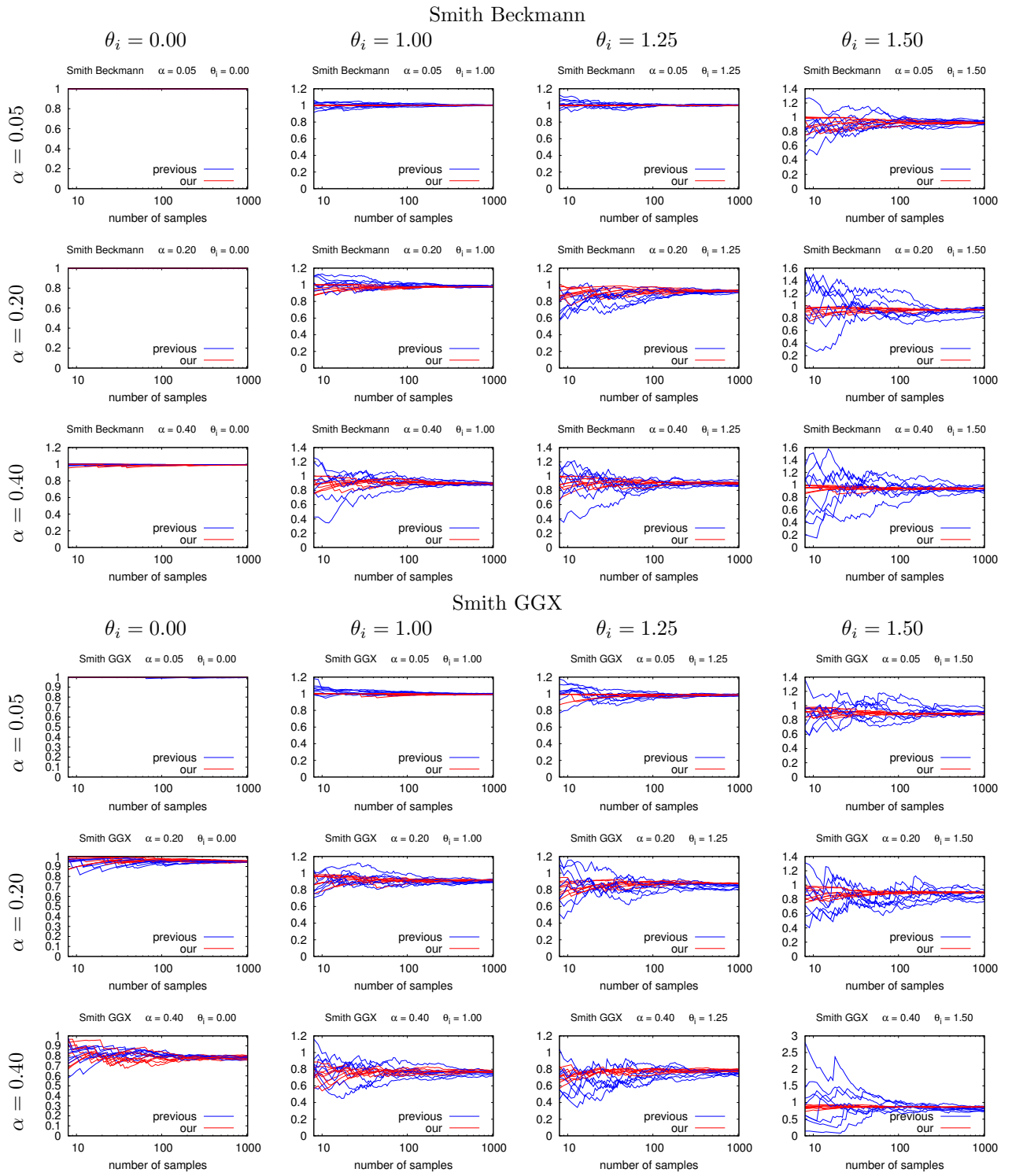
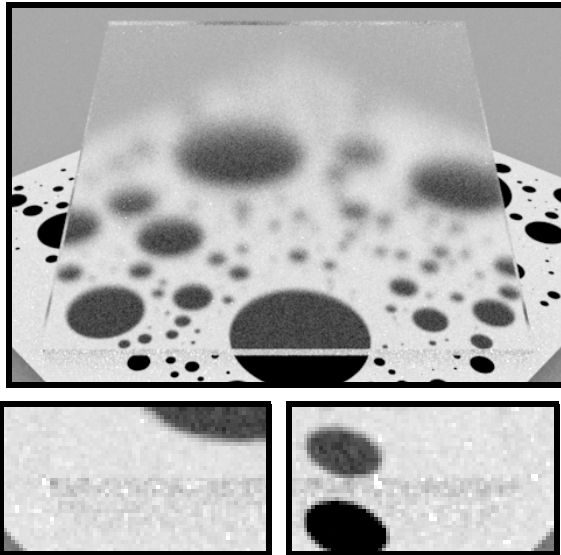


FIGURE F.5 – Importance sampling with the Smith model.

F.6 Results

F.6.1 Isotropic Beckmann

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (11.4s)



Our : BSDF Importance sampling using the distribution of visible normals. 58 spp (11.4s)

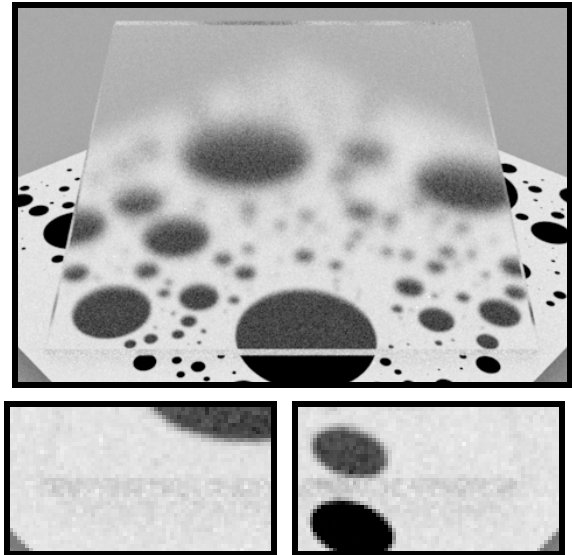
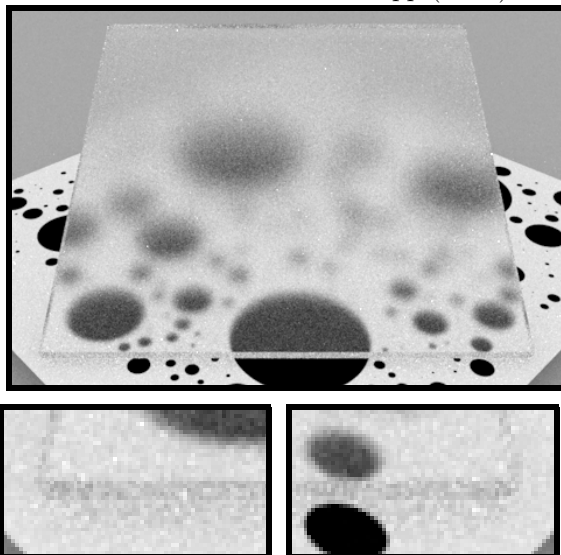


FIGURE F.6 – A dielectric glass plate ($n = 1.5$) with isotropic Beckmann roughness ($\alpha = 0.1$) on all faces (with the Smith masking function).

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (11.5s)



Our : BSDF Importance sampling using the distribution of visible normals. 58 spp (12.1s)

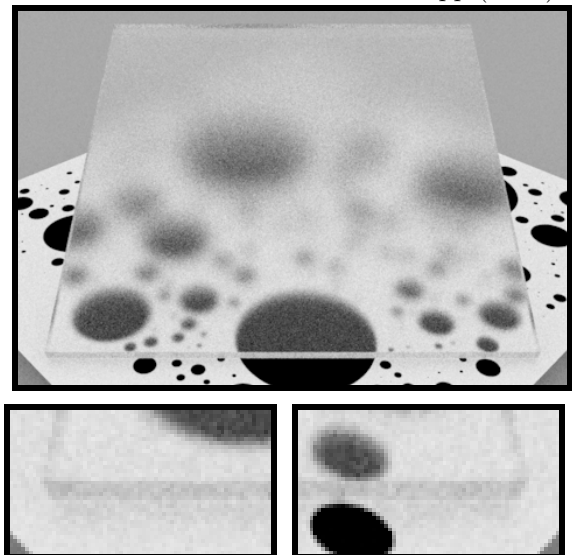
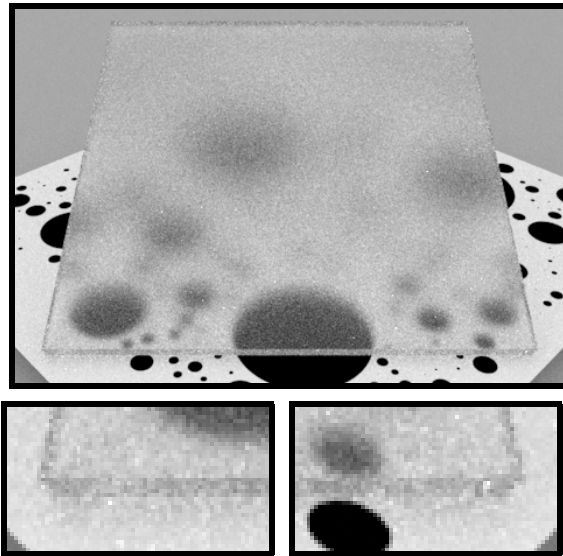


FIGURE F.7 – A dielectric glass plate ($n = 1.5$) with isotropic Beckmann roughness ($\alpha = 0.2$) on all faces (with the Smith masking function).

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (11.1s)



Our : BSDF Importance sampling using the distribution of visible normals. 52 spp (11.0s)

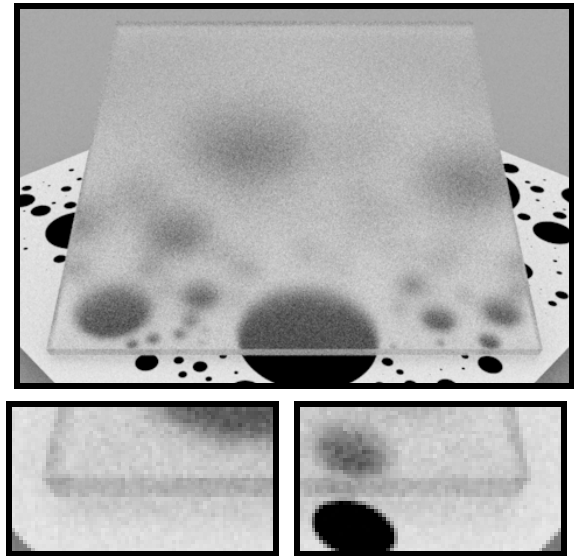
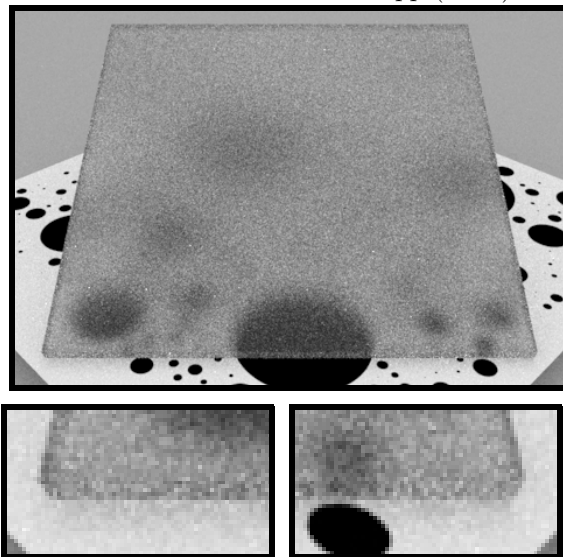


FIGURE F.8 – A dielectric glass plate ($n = 1.5$) with isotropic Beckmann roughness ($\alpha = 0.4$) on all faces (with the Smith masking function).

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (10.7s)



Our : BSDF Importance sampling using the distribution of visible normals. 50 spp (10.4s)

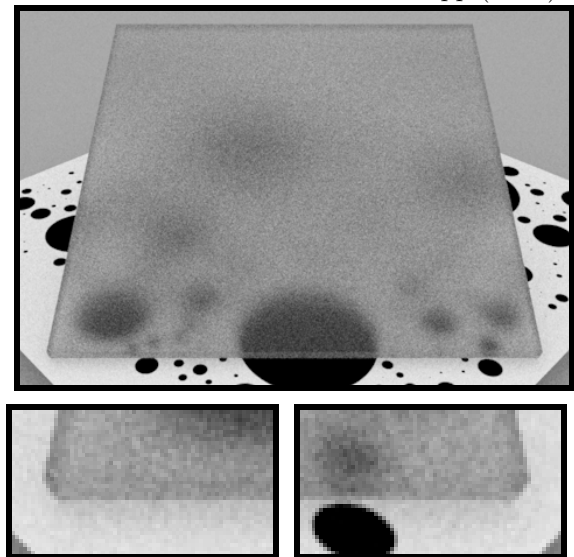
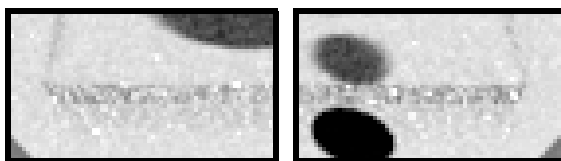
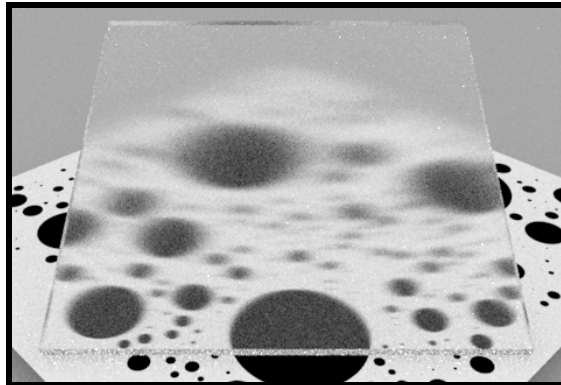


FIGURE F.9 – A dielectric glass plate ($n = 1.5$) with isotropic Beckmann roughness ($\alpha = 0.8$) on all faces (with the Smith masking function).

F.6.2 Anisotropic Beckmann

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (12.3s)



Our : BSDF Importance sampling using the distribution of visible normals. 58 spp (13.4s)

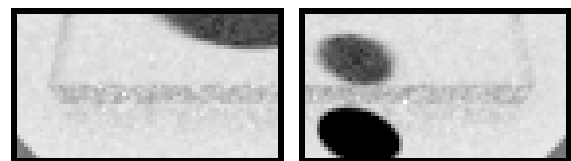
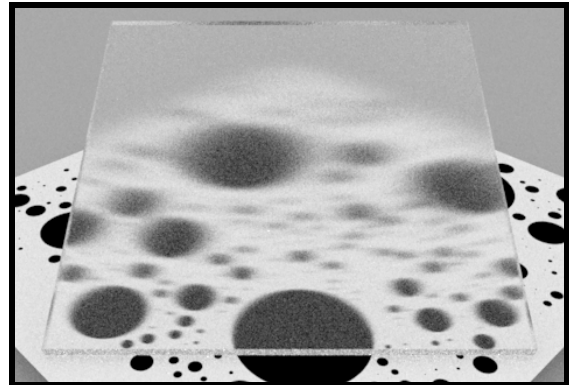
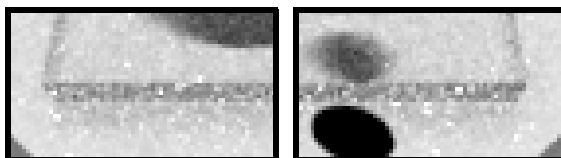
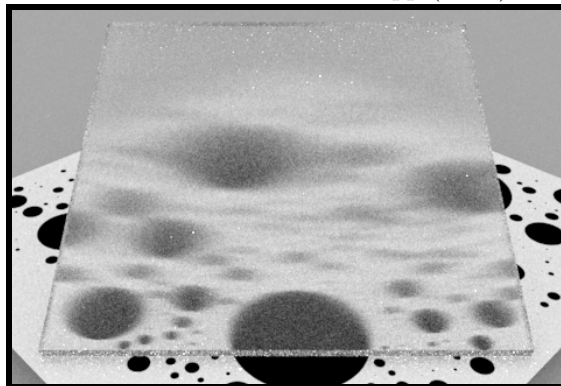


FIGURE F.10 – A dielectric glass plate ($n = 1.5$) with anisotropic Beckmann roughness ($\alpha_x = 0.03$, $\alpha_y = 0.24$) on all faces (with the Smith masking function).

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (11.2s)



Our : BSDF Importance sampling using the distribution of visible normals. 54 spp (11.2s)

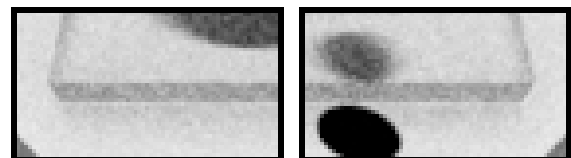
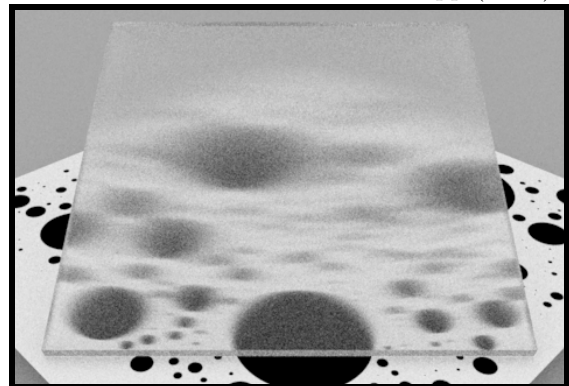
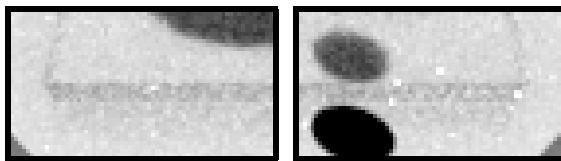
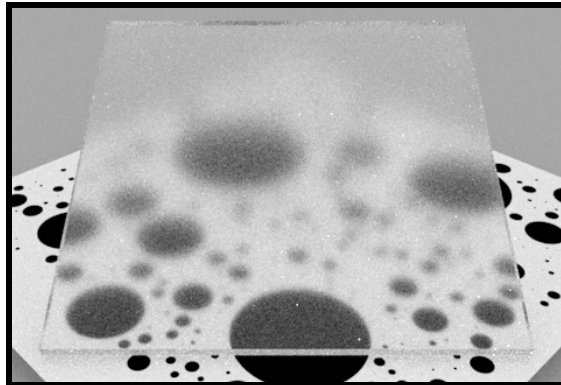


FIGURE F.11 – A dielectric glass plate ($n = 1.5$) with anisotropic Beckmann roughness ($\alpha_x = 0.03$, $\alpha_y = 0.48$) on all faces (with the Smith masking function).

F.6.3 Isotropic GGX

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (11.0s)



Our : BSDF Importance sampling using the distribution of visible normals. 58 spp (11.4s)

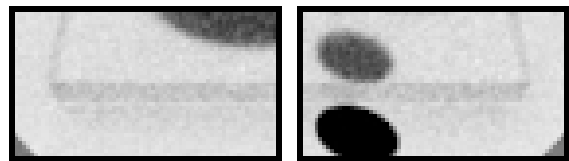
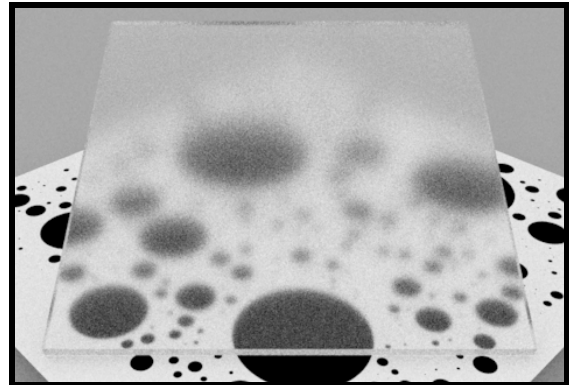
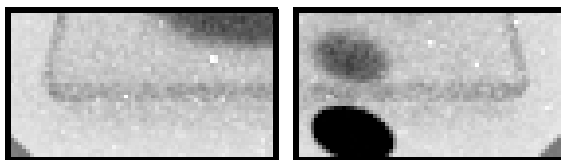
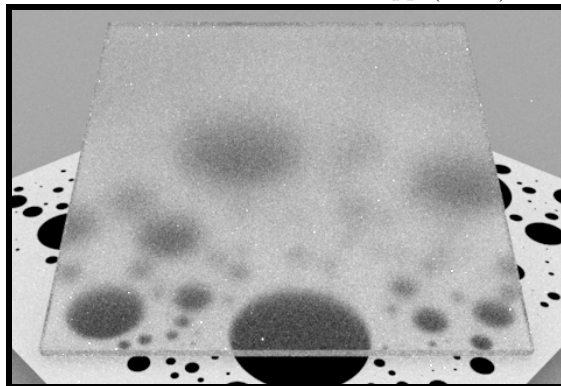


FIGURE F.12 – A dielectric glass plate ($n = 1.5$) with isotropic GGX roughness ($\alpha = 0.1$) on all faces (with the Smith masking function).

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (11.4s)



Our : BSDF Importance sampling using the distribution of visible normals. 58 spp (10.8s)

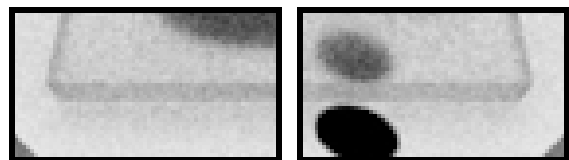
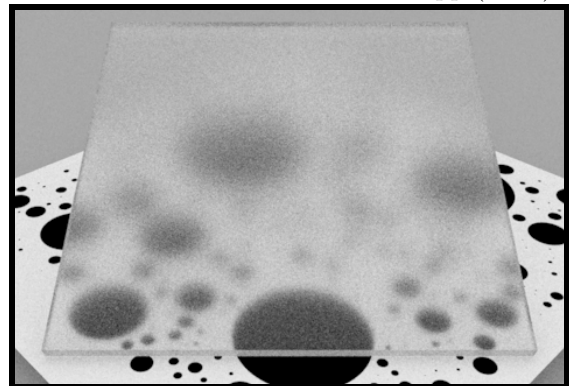
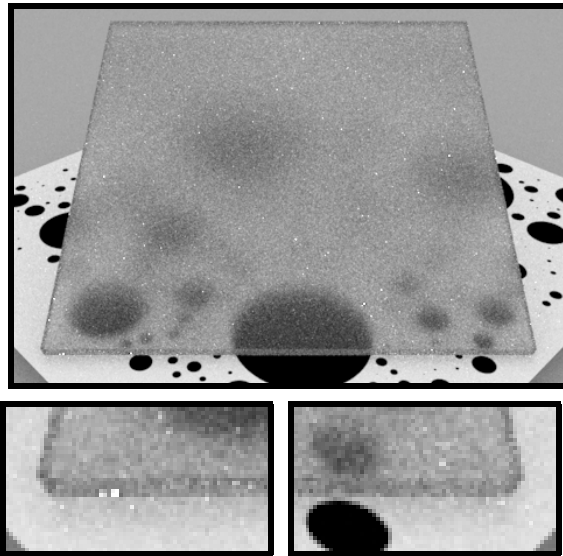


FIGURE F.13 – A dielectric glass plate ($n = 1.5$) with isotropic GGX roughness ($\alpha = 0.2$) on all faces (with the Smith masking function).

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (9.6s)



Our : BSDF Importance sampling using the distribution of visible normals. 56 spp (9.2s)

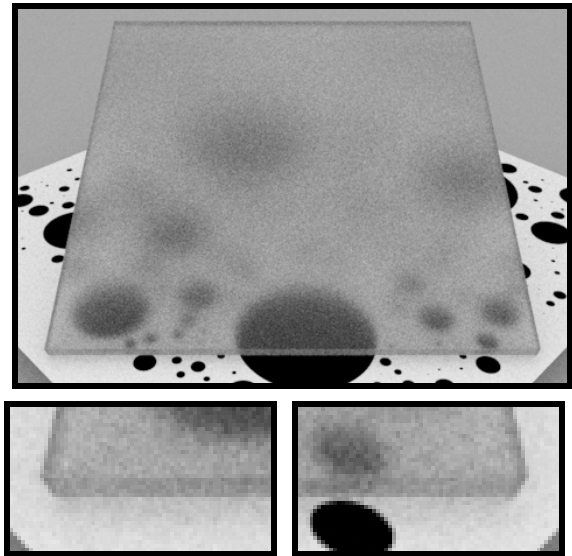
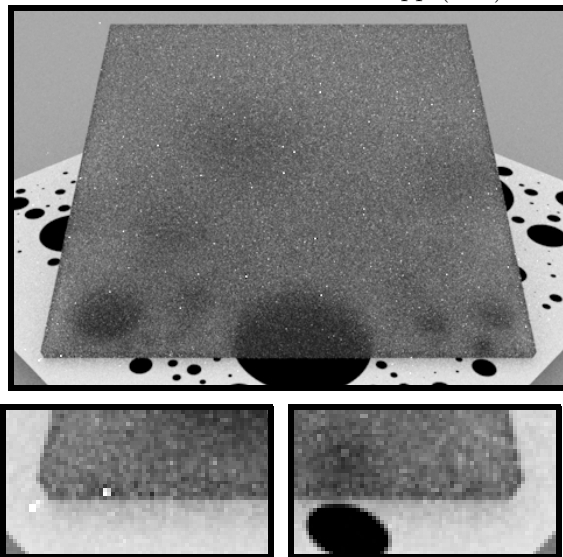


FIGURE F.14 – A dielectric glass plate ($n = 1.5$) with isotropic GGX roughness ($\alpha = 0.4$) on all faces (with the Smith masking function).

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (8.1s)



Our : BSDF Importance sampling using the distribution of visible normals. 52 spp (8.0s)

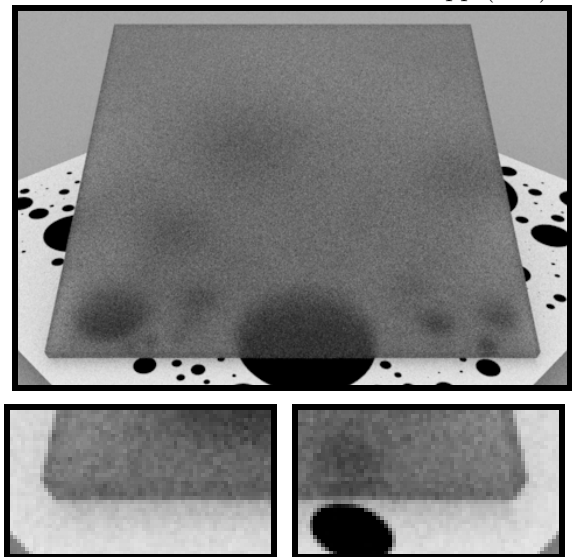
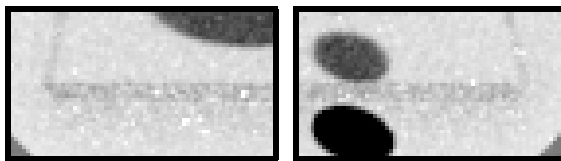
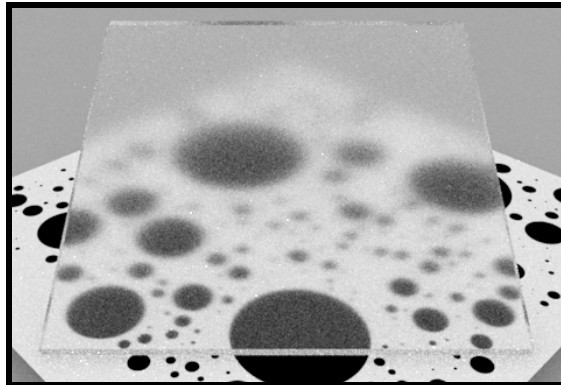


FIGURE F.15 – A dielectric glass plate ($n = 1.5$) with isotropic GGX roughness ($\alpha = 0.8$) on all faces (with the Smith masking function).

F.6.4 Anisotropic GGX

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (10.8s)



Our : BSDF Importance sampling using the distribution of visible normals. 58 spp (11.5s)

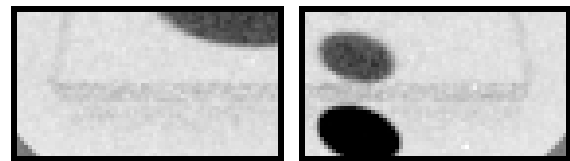
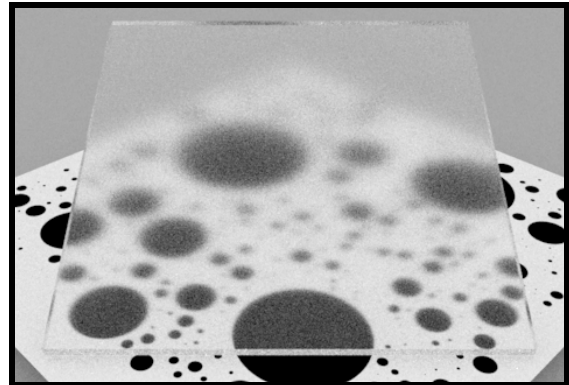
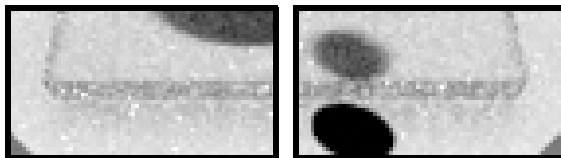
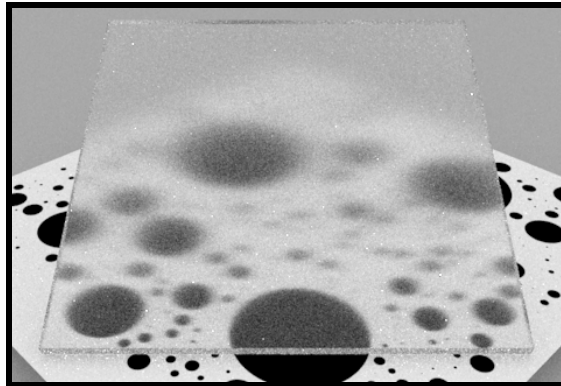


FIGURE F.16 – A dielectric glass plate ($n = 1.5$) with anisotropic GGX roughness ($\alpha_x = 0.05$, $\alpha_y = 0.10$) on all faces (with the Smith masking function).

Previous : BSDF Importance sampling using the distribution of normals. 64 spp (10.6s)



Our : BSDF Importance sampling using the distribution of visible normals. 52 spp (10.0s)

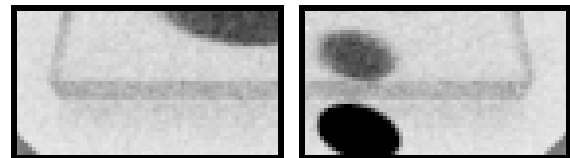
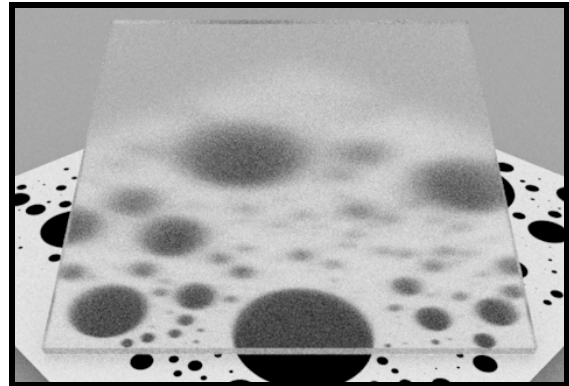


FIGURE F.17 – A dielectric glass plate ($n = 1.5$) with anisotropic GGX roughness ($\alpha_x = 0.05$, $\alpha_y = 0.20$) on all faces (with the Smith masking function).