



HAL
open science

Communication expressive de la forme au travers de l'éclairage et du rendu au trait

Romain Vergne

► **To cite this version:**

Romain Vergne. Communication expressive de la forme au travers de l'éclairage et du rendu au trait. Autre [cs.OH]. Université Sciences et Technologies - Bordeaux I, 2010. Français. NNT : . tel-00595068

HAL Id: tel-00595068

<https://theses.hal.science/tel-00595068>

Submitted on 23 May 2011

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

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

N° d'ordre : 4154

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET
D'INFORMATIQUE

par **Romain Vergne**

Pour obtenir le grade de

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

**Communication expressive de la forme au travers de l'éclairage et
du rendu au trait**

Soutenue le : 10 décembre 2010

Après avis des rapporteurs :

Joëlle Thollot Professeur
George Drettakis Directeur de recherche

Devant la commission d'examen composée de :

Joëlle Thollot	Professeur	Rapporteur
George Drettakis	Directeur de recherche		Rapporteur
Victor Ostromoukhov		Professeur	Examineur
Christophe Schlick	..	Professeur	Directeur de Thèse
Xavier Granier	Chargé de Recherche	.	Co-Directeur de Thèse
Pascal Barla	Chargé de Recherche	.	Co-Directeur de Thèse

Résumé

Le *rendu expressif* a pour objectif de créer des algorithmes permettant aux de contrôler la manière dont les images de synthèse sont créées. Plus précisément, il s'agit de laisser aux utilisateurs la possibilité de faire passer un message spécifique dans un style donné, même s'il faut pour cela s'éloigner des techniques d'éclairage physiquement réalistes. Dans cette thèse, nous proposons une formalisation du *rendu expressif*, en décomposant le message à communiquer dans une scène 3D et le style utilisé pour créer l'image finale. Au cours de ce manuscrit, nous nous concentrons sur deux domaines ayant la particularité de s'appuyer sur un message commun : la mise en valeur de la forme des objets dans des scènes dynamiques.

Dans la première partie de ce mémoire, nous proposons de nouvelles solutions pour exagérer les détails de surface à travers le rendu. Nous montrons notamment comment extraire des informations pertinentes sur la forme des objets en prenant les caractéristiques du système visuel humain en compte. Notre approche permet ainsi de déterminer des informations dépendantes du point de vue, qui fournissent des niveaux de détail automatiques de manière dynamique. Nous nous servons ensuite de ce message pour modifier l'éclairage et l'introduire dans des styles variés, allant du rendu minimaliste noir et blanc, au rendu réaliste.

Dans la seconde partie de ce mémoire, nous proposons une nouvelle approche pour communiquer la forme des objets via le dessin au trait. Nous introduisons une définition générale des lignes, cohérente spatialement et temporellement, et capable d'identifier de nombreux types de détails. Nous montrons ensuite comment appliquer une stylisation implicite le long de ces lignes, de sorte à obtenir des effets et styles originaux qui s'adaptent parfaitement aux détails de surface.

Nous discutons enfin plus en détail des limitations de chacune de ces approches dans une troisième partie. Nous montrons que notre perception de l'apparence est un élément essentiel sur lequel nous pourrions nous appuyer dans des travaux futurs. Les messages et styles dont nous parlons pourraient évoluer en un modèle plus intuitif et plus cohérent.

Mots-clés : Rendu expressif, Rendu non-photoréaliste, Communication de la forme, Dessin au trait.

Abstract

Expressive rendering aims at designing algorithms that allow to control the way images are created. More precisely, it provides the possibility to convey a specific message using a given style, even if non-physically based techniques have to be used. In this thesis, we propose a formalization for *expressive rendering*, by decomposing it into a message to convey in a 3D scene, and the style used to create the final image. We will concentrate our work on two topics that have a same message : object shape enhancement on dynamic scenes.

In the first part of this thesis, we propose new solutions for exaggerating surface details through shading. We show how to extract relevant surface features by taking the human visual system into account. Our approach then allows to detect view-dependant surface details and provides automatic levels-of-detail on dynamic scenes. This message is used for modifying the way lights interact with 3D objects and for creating various styles, from minimal black and white toon shading to realistic rendering.

In the second part of this thesis, we propose a new approach for conveying surface details using line-based rendering. We show that all kind of lines may be extracted via a single general definition, in a temporally and spatially coherent way. Next, we show how to apply an implicit stylization along the detected features, in order to obtain various styles and original effects.

We discuss the limitations of both techniques in the third part of the thesis. We show that our perception of appearance is an essential element that have to be taken into account in future work. Messages and styles could evoluate in a more coherent and intuitive model.

Key-Words : Expressive rendering, Non-photorealistic rendering, Shape enhancement, Line-based rendering.

Remerciements

J'ai été exceptionnellement bien entouré durant toute la durée de cette thèse et je tiens à remercier toutes les personnes qui ont contribué à l'aboutissement de ces travaux. Je remercie tout particulièrement mes trois directeurs de thèse : Christophe Schlick, Xavier Granier et Pascal Barla. Toutes les idées contenues dans ce manuscrit proviennent de nos nombreuses discussions. Je remercie Christophe pour sa gentillesse et son incroyable faculté à trouver des solutions simples à des problèmes complexes. Je remercie Xavier, sans qui je n'aurais jamais commencé cette thèse. Il m'a soutenu du début à la fin (et continue de le faire) et a toujours tout fait pour me faire avancer. Je remercie aussi Pascal pour entre autres : sa multitude d'idées géniales, ses discussions (scientifiques ou non) autour de cafés, au fait qu'il ait été la tout le temps pour les bon et les moins bon moments, aux restos, aux deadlines, aux restos après les deadlines, et j'en passe.

Je souhaite aussi remercier tous les membres de mon jury. Mes deux rapporteurs de thèse, Joëlle Thollot et George Drettakis, pour leur intérêt envers les travaux présentés dans ce manuscrit, leur relecture et leurs remarques constructives. Je remercie Victor Ostromoukhov pour avoir accepté d'être examinateur lors de la soutenance et pour son intérêt envers mes travaux. Merci encore à Joëlle pour avoir accepté de présider le jury.

Je tiens à remercier Pascal Guitton qui m'a accepté au sein de l'équipe-projet IPARLA et qui m'a toujours soutenu, à chaque fois que j'en ai eu besoins. Je remercie plus généralement toute l'équipe IPARLA, dans laquelle j'ai passé trois années excellentes. Je m'y suis fait des amis, plus que des collègues. Je remercie notamment Romain Pacanowski, Jérôme Baril, Benoît Bossavit, Nicolas Mellado, Fabrice Declès, Martin Hachet, Patrick Reuter, Sebastian Knödel, Julien Hadim, David Vanderhaeghe pour les moments passés ensemble, les discussions (scientifiques ou pas), les cafés et les ping-pongs.

Je remercie toutes les personnes qui ont collaboré aux travaux présentés dans ce document et sans qui ceux-ci n'auraient jamais aboutis : Pascal Barla, Xavier Granier, Christophe Schlick, Romain Pacanowski, David Vanderhaeghe, Jiazhou Chen, Pascal Mora, Frédéric Boutouille.

Enfin, je tiens à remercier toute ma famille qui m'a toujours guidé et soutenu dans tout ce que j'ai pu entreprendre. Merci donc à mes parents que j'adore, à mes deux géniales soeurs, à mes beaux frères et à leurs enfants. Merci aussi à Sarah qui m'a supporté, notamment lors de la rédaction pendant laquelle je n'étais pas forcément disponible. Merci à tous mes amis, à toutes les personnes qui ont pris le temps de relire ce document et à tous ceux que j'ai oublié.

Table des matières

1	Introduction	1
2	Rendu expressif	5
2.1	Retour aux sources	5
2.1.1	Dessins artistiques	7
2.1.2	Illustrations scientifiques	9
2.2	Rendu expressif en informatique	11
2.3	Conclusion	18
I	Mise en valeur de la forme à travers l'éclairage	19
3	L'éclairage et la forme	21
3.1	Réintroduction de la forme par les artistes	21
3.2	Équation de réflexion	24
3.2.1	Présentation de la fonction	24
3.2.2	BRDF	25
3.3	Les rendus stylisés existants	28
3.3.1	Styles <i>cartoon</i>	28
3.3.2	<i>Accessibility Shading</i> et <i>Ambient Occlusion</i>	30
3.3.3	Exagération des détails	31
4	Descripteurs de surface	35
4.1	Descripteur centré sur l'objet	35
4.2	Descripteurs hybrides	38
4.2.1	Courbure apparente [JDA07]	39
4.2.2	Descripteur de relief apparent	39
4.3	Descripteur centré sur le point de vue	43
4.3.1	Cas 1D	43
4.3.2	Analyse de la variation des normales en 2D	44

4.3.3	Sélection du niveau de détail	45
4.3.4	Implémentation	46
4.4	Comparaison des descripteurs	46
5	Modifications de l'équation de réflexion	49
5.1	Modification de l'intensité globale	50
5.1.1	Contrôle des formes avec une carte de relief apparent . . .	50
5.1.2	Styles	52
5.2	Modification des rayons lumineux	55
5.2.1	Motivations	55
5.2.2	Modification de l'équation de réflexion	56
5.2.3	Intégration dans différents styles	58
5.2.4	Comparaison avec les méthodes existantes	62
5.3	Modification des lumières réfléchies	63
5.3.1	Fonction de mise à l'échelle	64
5.3.2	Styles	66
5.4	Comparaison des approches	73
5.5	Discussion et travaux futurs	76
II	Représentation de la forme par le dessin au trait	79
6	La ligne et la forme	81
6.1	Les lignes créées par les artistes	81
6.2	Lignes définies à la surface des objets	85
6.3	Stylisation de lignes paramétrées	88
6.4	L'approche image	90
6.5	Une nouvelle vision du dessin au trait	91
7	Généralisation de la définition d'une ligne	93
7.1	Identification des lignes	94
7.1.1	Squelette des détails	94
7.1.2	Profil des détails	96
7.2	Implémentation	97
7.2.1	Extraction des données	97
7.2.2	Échantillonnage du profil	98
7.2.3	<i>Fitting</i> du profil	100
8	Stylisation de la forme via des brosses implicites	103
8.1	Convolution basée sur les détails de surface	104
8.2	Applications	107
8.3	Comparaison des stylisations	110
8.4	Discussion	111

III Discussions et conclusion	113
9 Vers un nouveau modèle d'apparence	115
9.1 Observations	115
9.1.1 Forme, lumière, matériau	115
9.1.2 Variations de forme VS variations d'intensités lumineuses	117
9.2 Besoins d'un nouveau modèle d'apparence	118
9.3 Deux angles d'attaque	121
9.3.1 Étude de l'équation de réflexion	121
9.3.2 Étude d'éléments perceptuels	121
10 Conclusion	125
10.1 Résumé des contributions	125
10.2 Nouveaux messages / nouveaux styles	126
Publications	129
Publications avec comité de lecture	129
Présentation	129
Publications sans comité de lecture	130
A Dérivation du jacobien	131

Introduction

L'infographie développe aujourd'hui des outils indispensables pour la production et la visualisation de données numériques. Les films, les dessins animés, les jeux vidéo, la publicité, l'art, la médecine, la biologie, les mathématiques, la physique, l'astronomie, l'archéologie représentent, entre autres, autant de secteurs dans lesquels l'utilisation et les besoins en outils graphiques sont de plus en plus grands. Il est aujourd'hui souvent plus facile de manipuler des images en utilisant des techniques infographiques plutôt que de les modifier à la main. C'est le cas lors du tournage d'un film, où il est plus efficace d'ajouter ou d'enlever des éléments d'une scène en post-production plutôt que de trouver un décor adapté au tournage. Dans toutes les applications, les dessinateurs et artistes ont besoin de pouvoir créer et manipuler des styles identifiables. Que ce soit un rendu cartoon dans un jeu vidéo, un rendu réaliste pour un film, ou un rendu au trait dans une illustration (Figure 1.1), il est nécessaire de donner des outils adaptés à l'utilisateur. Malgré leur très grande versatilité, ces styles numériques restent difficiles à contrôler. Prenons l'exemple d'une scène 3D dans un film d'animation. Dans un style



Figure 1.1: Les styles permettent de créer une ambiance et de faire passer des informations spécifiques. À gauche : le rendu *cartoon* du jeu vidéo *Queen of the Holy White Ash* (©Ninokuni) donne un style très clair et épuré. Au milieu : la teinte et la saturation des nuages dans le film *300* (©Warner Bros) permettent d'accentuer la gravité de la scène. À droite : le crayon utilisé dans cette illustration scientifique permet de montrer tous les détails du véhicule (©René Dosne).

réaliste, les artistes commencent par créer un environnement lumineux en plaçant, dans un logiciel de modélisation 3D, les différentes lumières dont ils vont avoir besoin. Ensuite, ils appliquent un algorithme d'éclairage global sur les différents éléments de la scène. Il s'agit d'une technique physiquement réaliste simulant les interactions complexes qui se produisent entre la forme des objets, leurs matériaux et les rayons lumineux. Cela peut prendre quelques heures avec une ferme de calcul pour obtenir un résultat de qualité. Une fois ces images créées, elles sont placées dans un logiciel de composition pour effectuer les dernières retouches : adaptation de tons, modification des couleurs, combinaison de différentes couches, etc. Si un problème survient à cette étape de la chaîne, il faut identifier parmi tous les paramètres résultants ceux qu'il faut modifier, et recommencer les étapes correspondantes. Cela peut arriver par exemple, si la forme des objets n'est pas assez visible dans le rendu final, ou si les reflets lumineux ou les ombres ne sont pas bien placés. Les studios d'effet spéciaux tentent de limiter au maximum ces aller-retour. Pour cela, ils exportent les rendus sous la forme de nombreux calques (avoisinant parfois la centaine) pour ensuite les recombinaison.

D'autres problèmes se posent dans les applications interactives. Dans ce cas, le temps réduit de calcul joue un rôle important. Il faut pouvoir générer les images très rapidement (de l'ordre d'une image toutes les $1/25^{\text{ième}}$ de seconde) quel que soit le style choisi. Des précalculs sont parfois utilisés pour stocker des informations concernant le rendu dans des textures ou dans des formules. C'est ce qui est fait dans les jeux vidéo par exemple. Néanmoins, une partie dynamique est toujours nécessaire, notamment lorsque le point de vue entre en compte. Dans tous les cas, il faut pouvoir laisser à l'utilisateur la possibilité de créer son propre style afin de faire passer un message spécifique (expression d'un personnage, relations spatiales entre les objets, etc.).

Le domaine du *rendu expressif* (issu du Rendu Non-Photoréaliste ou *NPR*¹) a pour objectif de résoudre ces problèmes. Plus précisément, il s'agit de créer des algorithmes permettant à l'utilisateur de faire passer un message qui lui est propre, même s'il faut pour cela s'éloigner des techniques physiquement réalistes. Le *rendu expressif* est un domaine vaste, utilisé dans toutes les applications et utilisant tout type d'entrées (images, vidéos, scènes 3D, etc.). Il va permettre de créer une ambiance précise en simulant, par exemple, certaines techniques classiques comme la peinture ou les dessins, mais surtout de faire passer une information essentielle et améliorer la lisibilité de l'image créée. Un des exemples les plus connus de *rendu expressif* est la carte routière (Figure 1.2). Il est bien plus facile de se repérer dans l'espace en utilisant des informations épurées, avec des abstractions sur le paysage, des indications de lieux, et des types de lignes différentes pour les routes plutôt que de se baser sur une photographie satellite.

¹Pour *Non-Photorealistic Rendering* en anglais.

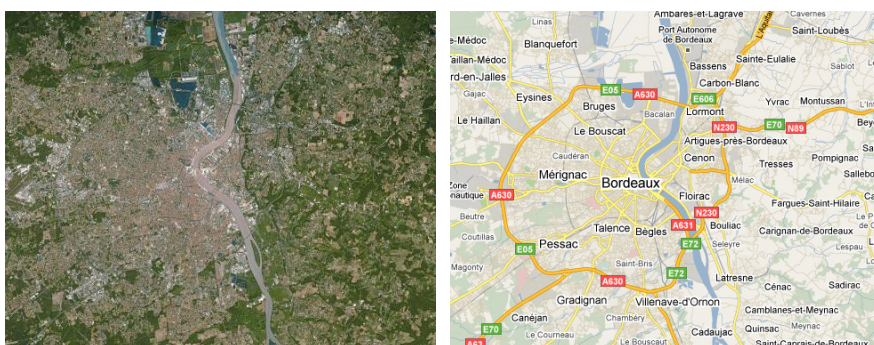


Figure 1.2: Par comparaison avec une photographie satellite, la carte routière a été conçue de manière à montrer des éléments indispensables pour se repérer dans l'espace : les couleurs spécifiques des routes, leurs directions, certains lieux, etc. ©Google maps.

Dans cette thèse, nous proposons une formalisation du *rendu expressif* pour l'informatique, en décomposant le *message* à communiquer dans une scène 3D et le *style* utilisé pour créer le rendu final (cf. Figure 1.3). Le message contient une description de la scène permettant d'envoyer les informations ou paramètres nécessaires au style. Le style, quant à lui, contraint le message pour que celui-ci ne lui donne que des paramètres facilement représentables.

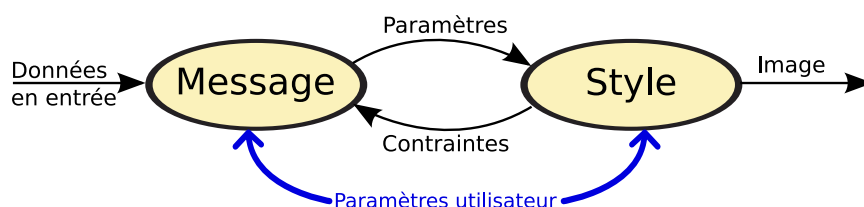


Figure 1.3: Message et Style.

Une telle formalisation est nécessaire en informatique graphique ; si les artistes peuvent dessiner l'apparence avec l'utilisation directe d'outils qui contraignent le style, nous ne disposons que d'informations physiques lorsque l'on s'intéresse à des données 3D : les maillages des objets, les environnements lumineux, les matériaux, les caractéristiques de la caméra, etc. Nous discuterons de ces différences dans le Chapitre 2 : nous examinerons dans un premier temps comment les artistes parviennent à communiquer des ambiances et des informations avec des styles spécifiques, puis nous replacerons quelques exemples de *rendu expressif* dans ce contexte pour distinguer concrètement les messages, les styles et leurs interactions.

Nous nous concentrerons ensuite sur deux domaines de *rendu expressif* ayant la particularité de s'appuyer sur un message commun : la mise en valeur de la forme des objets dans des scènes 3D dynamiques. Les applications dans lesquelles ces travaux s'inscrivent sont très vastes, allant du jeu vidéo aux films d'animation en passant par la visualisation scientifique.

Plus précisément, nous proposons de nouvelles solutions pour exagérer les détails de surface à travers le rendu dans la première partie. Nous montrons notamment comment extraire des informations de forme dépendantes du point de vue en nous basant sur les caractéristiques du système visuel humain dans le Chapitre 4. Nous nous servons ensuite de ce message pour modifier l'éclairage et l'introduire dans des styles variés, allant du rendu NPR minimaliste au rendu réaliste, dans le Chapitre 5.

Dans la seconde partie, le rendu au trait sera étudié plus précisément car il permet de communiquer la forme avec une économie de moyens (cf. Chapitre 6). Nous proposons tout d'abord une nouvelle manière d'extraire les lignes dans le Chapitre 7. Il s'agit d'une définition générale, capable d'extraire de nombreux types de détails et qui reste cohérente à la fois spatialement et temporellement sur des scènes dynamiques. Nous montrons ensuite comment appliquer une stylisation implicite le long de ces lignes, de sorte à obtenir des effets qui s'adaptent parfaitement aux détails de surface dans le Chapitre 8.

Nous discuterons enfin plus en détail des limitations de chacune de ces approches. Nous verrons que notre perception de l'apparence est un élément essentiel sur lequel nous pourrions nous appuyer dans des travaux futurs. Les messages et les styles dont nous parlerons pourraient évoluer en un modèle plus intuitif et plus cohérent.

Rendu expressif

2.1 Retour aux sources

Les artistes n'ont pas attendu la venue de l'informatique pour créer des images expressives. Depuis toujours, les hommes dessinent, imaginent de nouveaux styles, utilisent de nouveaux outils dans le but d'obtenir des résultats variés. Vue l'infinité de styles pouvant être produits, certaines personnes ont conçu des espaces dans lesquels il est plus facile de se repérer, de manière à comparer les œuvres et se faire une idée de leurs différences.

C'est le cas de Scott McCloud [McC94] qui a tenté de rassembler tout le vocabulaire concernant les dessins dans un schéma représentant l'ensemble des possibilités (Figure 2.1 (a)). Son idée est de représenter toutes les formes d'art visuel sur un triangle. Dans le coin inférieur gauche sont placées toutes les créations artistiques réalistes. En se déplaçant vers la droite, des niveaux d'abstraction iconiques sont introduits. On trouve alors dans le coin inférieur droit les dessins qui communiquent un sens, dans lesquels des détails ont été enlevés ou exagérés, et où les formes ont été conceptualisées, jusqu'à arriver aux mots du langage. En partant vers le sommet du triangle, ce sont des abstractions pures qui sont identifiées. On s'éloigne ainsi de la réalité et on se rapproche du plan pictural dans lequel se trouvent les figures géométriques simples, les couleurs et les lignes.

Il est parfois difficile de faire entrer certaines œuvres ou expressions artistiques dans le triangle de McCloud car il a été initialement conçu pour les dessinateurs et scénaristes de bandes dessinées. D'autres artistes ont modifié ce triangle pour le rendre plus général et/ou pour y différencier des thèmes. Par exemple, Goguely¹ a étendu ce triangle à un double tétraèdre (Figure 2.1 (b)) en distinguant notamment la vision naturelle et surnaturelle des œuvres avec les côtés abstraits et imaginaires.

¹<http://dendrographe.blogspot.com/>

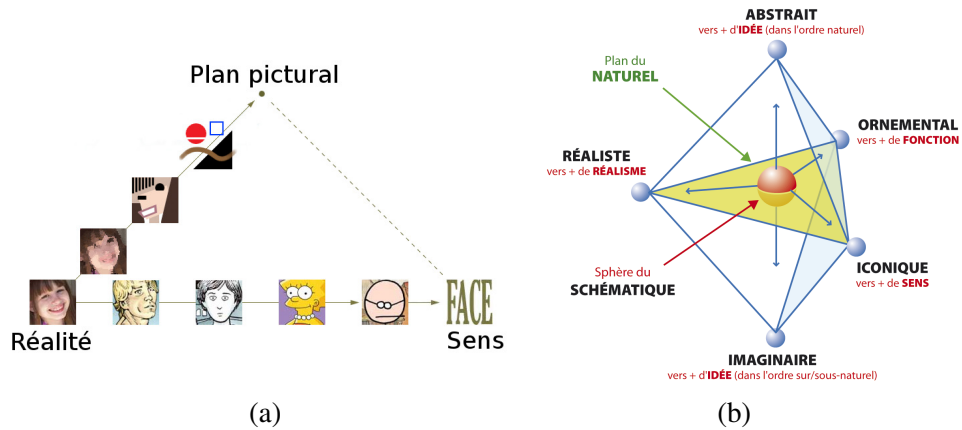


Fig. 2.1: (a) Le triangle de McCloud. (b) Le double tétraèdre des arts d'Yves Goguely.

Une vision plus formelle a été imaginée par Durand [Dur02] (à partir des travaux de Willats [Wil97]) dans laquelle il classe non pas les styles, mais les méthodes mises en œuvre pour créer les images. Il propose notamment de séparer les techniques de dessin en quatre problématiques :

- Le système spatial dont l'objectif est de transformer les données originales dans le plan image. L'artiste décide ici quel type de projection sera utilisé (orthographique, perspective, curviligne, etc.).
- Le système de primitives concerne les manières de transformer les primitives 3D (points, lignes, surfaces, volumes) en espace image (points, lignes, régions).
- Le système d'attributs s'occupe d'affecter des propriétés visuelles à chacune des primitives, comme la couleur, la texture, la transparence ou l'orientation.
- Le système de marques correspond à la manière de styliser une primitive en utilisant ses attributs. L'artiste peut appliquer des points, des traits ou tout autre style pour représenter les primitives.

Bien entendu, il existe des relations fortes entre ces quatre systèmes car il est très complexe de décomposer une œuvre. Par exemple, une variation de couleur dans un coup de pinceau peut être définie par le système d'attributs, le système de marques ou les deux.

Dans cette thèse, nous ne recherchons pas de nouvelle classification, mais nous étudions ce qu'il y a en commun dans ces catégories. Notre approche est de considérer que toute composition artistique contient un certain **message** destiné au spectateur ou à l'utilisateur, et que ce message est mis en œuvre avec un **style** spécifique. Tout comme les lézards imbriqués que l'on peut voir dans l'œuvre d'Escher (cf. Figure 2.2), le message et le style sont très fortement liés. Si par exemple, l'objectif

d'un dessin est de représenter les détails les plus fins d'un objet, il sera beaucoup plus facile pour l'illustrateur de choisir un crayon plutôt que d'utiliser l'aquarelle. Cette dernière est plus généralement employée pour créer des aplats et des effets de couleur dans le rendu. Le choix d'un message, qui consiste dans cet exemple à montrer les détails, contraint donc le choix de l'outil et du style (un crayon fin ici). Inversement, le style impose des contraintes sur le message : il sera impossible de dessiner des détails plus fins que ce que la mine du crayon le permet. Les sections qui suivent montrent que ces notions simples sont présentes à un niveau plus complexe dans tous les dessins, que ce soient des peintures impressionnistes ou des illustrations scientifiques.



Fig. 2.2: Le **message**, le **style** et les liens qui les unissent sont représentés par deux lézards imbriqués l'un dans l'autre (©Escher).

2.1.1 Dessins artistiques

Les dessins ou peintures artistiques sont généralement créés dans le but de raconter une histoire ou de faire passer une sensation au spectateur. Les artistes jouent sur les couleurs, les formes, la disposition des objets dans la scène pour créer une image racontant un événement, décrivant les caractéristiques d'une personne ou d'un objet et faire passer des émotions diverses au spectateur. La série de tableaux de la Figure 2.3 montre à quel point le style joue un rôle sur le message et les émotions que les artistes ont voulu transmettre.

Géricault n'a par exemple pas choisi un style réaliste de manière anodine lorsqu'il a peint "Le radeau de la méduse". Il voulait raconter une histoire vraie, la désolation mais aussi l'espoir que vivaient ces hommes après le naufrage de leur frégate. Il a même étudié des cadavres en détail avant de dessiner les corps avec des coups de pinceau denses et des couleurs sobres et pâles. L'atmosphère pesante et les horreurs qu'ont vécues ces hommes à bord du radeau (comme le cannibalisme) sont non-seulement représentées par les gestes expressifs de ceux-ci, mais aussi avec les couleurs sombres du ciel, les ombres des nuages et l'océan qui s'agite.



F . 2.3: (a) Le radeau de la méduse - Théodore Géricault, 1819. (b) Autoportrait - Van Gogh, 1889. (c) Le cri - Edvard Munch, 1893. (d) Carré noir sur fond blanc - Kasimir Malevitch, 1930.

Dans l’“autoportrait” de Van Gogh, le style impressionniste utilisé tente de transmettre l’état d’âme complexe et mystérieux du peintre, interné à l’hôpital Saint Rémy à cette époque. En utilisant seulement différentes tonalités de bleu (mis à part la barbe rousse), Van Gogh montre son inquiétude et sa souffrance intérieure. Les effets d’ondulation des coups de pinceaux sur le fond renforcent le sentiment de malaise et les tourments psychologiques de l’artiste.

“Le cri” d’Edvard Munch montre comment les artistes s’éloignent de la réalité à l’époque de l’art nouveau, de manière à faire passer un message plus abstrait, une sensation. Dans ce tableau, les lignes fluides disposées en arc de cercle ont pour but d’incarner le cours de la pensée. Le personnage central, représenté avec un visage en forme de crâne et des couleurs qui s’éloignent de la réalité, montre l’angoisse et les peurs de l’homme moderne. Le mouvement ondoyant de la nature et la couleur rouge sang du ciel renforcent cette émotion.

Dans “Le carré noir sur fond blanc”, le suprématisme de Kasimir Malevitch atteint un paroxysme dans l’abstraction. On peut voir ici que le style contraint le message d’une manière absolue. Et c’est d’ailleurs ce que voulait montrer l’artiste. Il a renoncé à la couleur, à la perspective, et même à la forme en ne dessinant qu’un simple carré, de manière à libérer l’art de l’objectivité. Il souhaitait exprimer les lois internes de la peinture en enlevant toute identification à des sujets précis comme des figures ou des paysages.

2.1.2 Illustrations scientifiques

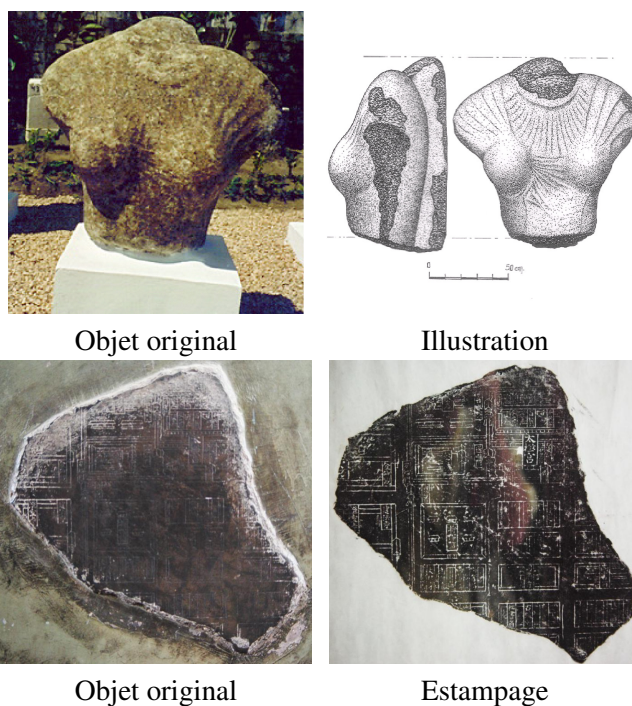


Fig. 2.4: Deux styles différents sont montrés ici pour mettre en valeur les détails sur des objets d’étude archéologiques. En haut : buste d’Isis (Alexandrie - Égypte) ©Isabelle Hairy. En bas : forêt de pierres (Xian - Chine). Photographies prises par Xavier Granier.

Contrairement aux dessins artistiques dont le but est de faire passer une émotion, une sensation ou une idée, les illustrations scientifiques ont pour objectif de rendre une représentation plus lisible, en accentuant ou en enlevant des détails, et en effectuant parfois des abstractions sur la forme. Le message à communiquer dans une illustration scientifique est donc souvent plus concret et peut concerner de nombreuses caractéristiques. Souvent, il s’agit de mettre en valeur la forme et les détails des objets. La Figure 2.4 montre le dessin réalisé par un archéologue du

buste d’Isis. Cette illustration montre à quel point le dessin met les détails de forme en valeur si on compare avec la photographie. Des pointillés sont utilisés pour le style : ils permettent de dessiner les nombreux détails, mais aussi de représenter les contrastes de manière efficace et sont bien adaptés à l’impression. Sur la ligne du bas, on peut voir un estampage réalisé dans le but de montrer les détails d’une pierre gravée. Cette méthode consiste à “calquer” les creux et les bosses d’une gravure sur une feuille à l’aide d’une brosse pour obtenir un négatif proche de l’objet original. Dans cet exemple, la méthode contraint le style de manière drastique. Seules deux couleurs sont utilisées : le noir pour les convexités et le blanc pour les concavités.

Si les détails de forme représentent souvent une caractéristique importante, il se peut aussi que l’illustrateur veuille mettre l’accent sur d’autres aspects. Le mouvement, le matériau, ou la relation spatiale entre plusieurs objets sont d’autres attributs parfois essentiels à la visualisation. Dans son illustration inspirée de Léonard de Vinci et destinée à ses étudiants (cf. Figure 2.5 (a)), Caryn Babian met en valeur plusieurs aspects du corps humain à la fois, comme la disposition et les relations entre les organes, la transparence, les rotations des membres et les sections transverses. Malgré le style très contraint qui dans ce cas est la craie sur un tableau noir, le professeur montre de nombreux détails sans que le résultat soit illisible. La seconde illustration montre comment on peut aussi vouloir mettre en valeur le matériau dont les objets sont composés. Pour montrer la transparence de ce bocal situé sur un fond noir, des intensités plus claires sont employées sur les régions contenant des reflets et sur les zones inclinées.

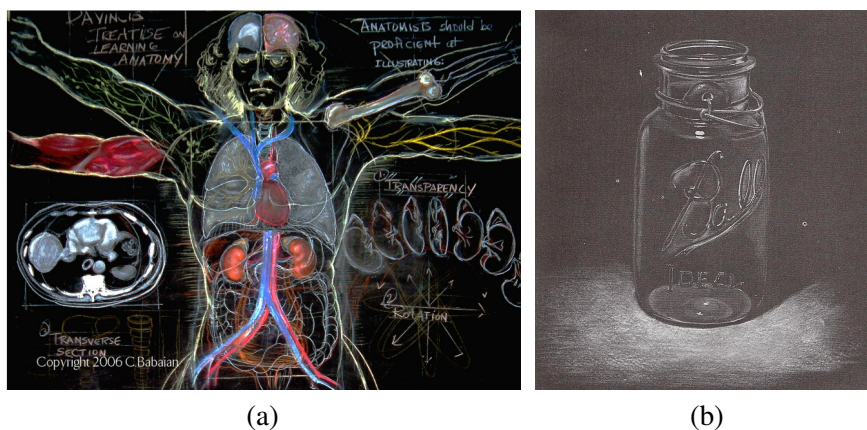


Figure 2.5: (a) Illustration de Caryn Babian primée par la revue “Science” lors du challenge de la visualisation scientifique en 2006. (b) Illustration tirée de [Woo94].

2.2 Rendu expressif en informatique

Le *rendu expressif* est un domaine récent qui a beaucoup évolué durant ces vingt dernières années. Son premier objectif a été d’imiter des styles artistiques existants, ou d’en créer de nouveaux grâce aux moyens offerts par les ordinateurs. C’est la raison pour laquelle ce domaine a tout d’abord été appelé “Rendu Non Photoréaliste” : il concernait toutes les techniques de dessin ou d’éclairage qui ne prenaient pas les lois de la physique en compte pour afficher une scène, contrairement aux méthodes d’éclairage global. Depuis lors, le but de ce domaine n’a plus été seulement de reproduire des styles artistiques existants, mais aussi de s’attaquer aux illustrations scientifiques, et plus généralement, de communiquer un message spécifique en se basant sur un style donné. Comme exprimé dans [Dur02], ce terme n’est pas bien défini car il a été choisi comme l’opposé d’un autre domaine. On peut d’ailleurs se demander à quel point les images ou les photographies créées sont elles même réalistes. Elles représentent en effet déjà une réalité altérée par des choix humains (cadrage, exposition, colorimétrie, profondeur de champs, objectifs, filtres, etc.). Parmi les quatre photographies présentes dans la Figure 2.6, on peut se demander quelle est celle qui se rapproche le plus des lois de la physique. Où se trouve la limite entre “réaliste” et “non réaliste” ? L’exemple de la Figure 2.7 montre une image générée avec un algorithme physiquement réaliste. Pourtant, il s’agit bien d’un film d’animation qui n’a absolument rien de réaliste. La modélisation et l’animation entrent aussi en compte dans le rendu d’une image. Dans cette thèse, nous considérerons donc que le rendu photoréaliste est un style comme un autre qui a pour point de départ les lois de la physique, mais qui a aussi pour but de communiquer un message.



Fig. 2.6: Laquelle de ces photographies est la plus “phtoréaliste” ?
Crypte de la cathédrale Saint-Denis ©Xavier Granier.



Figure 2.7: Image non réaliste générée à partir d’algorithmes physiquement réalistes. Shrek 4 ©PDI Dreamworks.

La plupart des méthodes précédentes de *rendu expressif* peuvent être décrites en termes de message et de style (cf. Figure 1.3). Par la suite, nous nous intéressons à quelques exemples pour les replacer dans ce contexte et y distinguer concrètement les messages, les styles et leurs interactions.

Les hachures ont par exemple fait l’objet de nombreuses recherches dans le domaine du *rendu expressif*. Les traits qui se chevauchent représentent la particularité de ce style. Dans les bandes dessinées ou les illustrations scientifiques, il permet de représenter le volume des objets ainsi que l’intensité de la lumière reflétée (le ton). Elles ont par exemple été utilisées dans [SABS94, WS94, SALS96, WS96, SWHS97, FMS02] pour reproduire le style de dessin fait au crayon et à la plume sur des images et des objets 3D. Ces articles traitent principalement du problème d’éclairage. Selon que les régions à styliser sont claires ou foncées, les hachures engendrées doivent être plus ou moins longues, épaisses et denses. Le principe général est d’utiliser une série de textures composées d’un nombre de traits plus ou moins important suivant que l’on veuille obtenir une zone claire ou foncée. Ces textures peuvent aussi être orientées de manière à ce que les hachures suivent des directions privilégiées qui représentent la forme de la surface. Les premières méthodes interactives pour créer des hachures utilisent des niveaux de détail pour chacune des textures comme les *Tonal Art Maps* [PHWF01, WPFH02, Fun03] qui permettent d’obtenir un rendu dans lequel le nombre de traits utilisés reste cohérent, que l’objet 3D soit proche ou loin de la caméra (cf. Figure 2.8). Dans tous ces papiers, le contrôle du placement des lignes et leur stylisation reste néanmoins limité. Ostromoukhov [Ost99] montre par exemple comment combiner les hachures de manière très propre pour reproduire le style de gravure sur cuivre. La paramétrisation indispensable à leur placement est soit induite par l’utilisateur, soit calculée avec des méthodes complexes [PFH00]. Dans [HZ00, ZISS04], les hachures sont automatiquement orientées selon des directions basées sur les courbures principales. La forme des objets est alors mise en valeur et le style produit se rapproche

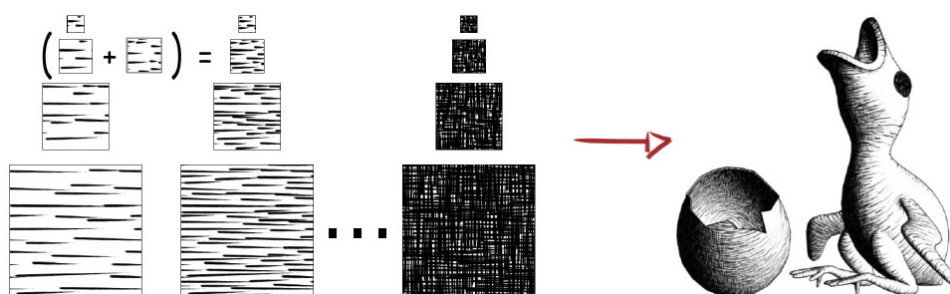


Figure 2.8: Les *Tonal Art Maps* (à gauche) sont plaquées sur un objet 3D (à droite) pour obtenir un rendu hachuré [PHWF01].

plus d’une illustration scientifique. Ces articles produisent les lignes qui suivent le flot avec un algorithme d’optimisation permettant de contrôler la distance qui les sépare [JL97]. Brunet *et al.* [BSSB99] placent aussi des lignes dans la direction des ombrages pour simuler un dessin au crayon à papier sur des modèles 3D. Plus récemment, des méthodes interactives ont vu le jour pour simuler le crayon à papier ou les hachures [LKL06, KYYL08]. Des textures sont automatiquement disposées sur des petits patches, puis mélangées intelligemment de telle sorte qu’aucune discontinuité n’apparaisse. Coconu *et al.* [CDH06] utilisent des primitives de plus haut niveau pour paramétrer et hachurer ainsi de la végétation.

Les pointillés ont aussi été largement étudiés en informatique. Comme pour les hachures, le message consiste ici à montrer efficacement les contrastes qu’il peut y avoir entre les zones d’ombres et les zones éclairées sur un objet ou dans une image. Ce genre d’affichage est aussi très utile pour l’impression. Si les méthodes citées précédemment sur les hachures peuvent aussi être utilisées pour ce style [Fun03], la répartition des points est essentielle pour communiquer efficacement l’intensité lumineuse. De nombreux articles utilisent une relaxation de Voronoï 2D pour disposer les points de telle sorte qu’ils soient uniformément distribués, sans pour autant être alignés [DHvOS00, Sec02, BSD09]. Kim *et al.* [KSL⁺08] ajoutent des contraintes pour que les points soient distribués le long du flot dans le but de créer le style bien spécifique du “Hedcut” (Figure 2.9 (a)). Celui-ci a été introduit par Kevin Sprouls pour dessiner des personnalités dans le “Wall Street Journal”. Des techniques de demi-ton [PQW⁺08, CAO09] utilisent aussi les points pour transformer des images en noir et blanc tout en conservant les détails de la scène. Pour avoir un contrôle plus précis sur la manière dont les points bougent lorsque les objets sont animés, des méthodes travaillant directement dans l’espace 3D ont été conçues. Par exemple, Baer *et al.* [BTBP07] utilisent une structure de données volumique et hiérarchique pour positionner les points et faire en sorte que l’intensité soit toujours constante lorsque les objets bougent ou s’éloignent de la caméra.



Fig. 2.9: (a) Style “Hedcut” utilisant des lignes et des pointillés pour représenter le visage de personnalités [KSL⁺08]. (b) Imitation de peinture à l’huile sur une image [ZZXZ10].

Les peintures à l’huile ont aussi pour objectif de positionner des marques sur la scène de sorte à reproduire des styles artistiques traditionnels comme l’impressionnisme, l’expressionnisme ou le pointillisme. Le message à faire passer est donc plus d’ordre visuel ; c’est la distribution des couleurs, l’orientation des coups de pinceaux autour des contours et la qualité de l’image rétinienne obtenue qui vont primer dans ce cas. Les coups de pinceaux d’une peinture à l’huile sont généralement représentés par des petites régions texturées. Si, comme pour les pointillés, la distribution des marques est importante, la principale difficulté réside dans le fait qu’il faut s’assurer que les marques recouvrent bien la totalité de l’image finale. Dans [Hae90], c’est l’utilisateur qui décide du positionnement des marques dans des images. L’une des premières méthodes automatiques a été développée par Litwinowicz [Lit97] : il utilise une grille régulière surimposée à l’image pour disposer les coups de pinceaux de sorte que toutes les zones soient traitées. Ceux-ci sont alors dessinés aléatoirement à l’intérieur de chaque case pour donner un aspect final non uniforme. [Her98] améliore la technique en itérant le procédé sur des images lissées. De grosses marques sont utilisées pour les images très floutées et leur taille diminue avec le niveau de détail. Dès lors, de nombreuses méthodes ont été mises en place [GCS02, CH02, SY00, HTER04, Her01, ZZ10, ZZXZ10] dans le but d’améliorer le positionnement des coups de pinceaux, mais aussi de simuler leurs formes et les interactions avec le support de manière efficace. Une imitation de pinceaux orientaux est par exemple appliquée dans [XLSN10]. Hertzmann [Her02] simule aussi les réflexions de la lumière sur les coups de pinceaux pour obtenir un résultat plus réaliste. Si le placement et le mélange des marques restent toujours un problème ouvert, la manière dont ils bougent dans les scènes animées est aussi très délicate à contrôler. Dans les vidéos, les marques peuvent

suivre le flot optique pour assurer la cohérence temporelle [HP00, LSF10], mais aussi s'appuyer sur des informations d'ordre plus sémantique et un découpage de la vidéo en plusieurs positions clés pour simplifier le problème [LZL⁺10]. Meier [Mei96] se place dans l'espace 3D et convertit les objets d'une scène sous la forme de particules pour les utiliser comme origines des coups de pinceaux. Placer les marques à la surface des objets 3D a l'avantage d'améliorer la cohérence temporelle car les primitives bougent de la même manière que les objets, mais cela enlève aussi certains effets 2D présents dans les peintures et les films d'animation. Vanderhaeghe *et al.* [VBTS07] relaxent certains de ces problèmes en permettant aux particules de bouger de manière cohérente tout en conservant une répartition correcte sur l'image.

Les aquarelles sont reproduites par les techniques infographiques en simulant les interactions entre pigments et papier. Elles se distinguent de la peinture car il s'agit non pas de produire des coups de pinceaux uniformément sur l'image, mais d'obtenir des aplats de couleurs possédant leurs propres caractéristiques. La variation de la saturation de la couleur le long des régions, la visibilité et la texture du papier sont en effet le résultat complexe de l'interaction entre l'eau, les pigments et le support. Encore une fois, l'objectif est de reproduire un style existant pour produire des effets visuels plaisants. Curtis *et al.* [CAS⁺97] obtiennent un rendu aquarelle physiquement réaliste en simulant les interactions du fluide avec le papier (cf. Figure 2.10 (a)). Cette méthode est néanmoins très longue à appliquer : il fallait environ sept heures pour produire une seule image à l'époque de cet article. De la même manière, Chu et Tai [CT05] simulent la dispersion de l'encre de Chine interactivement. Les algorithmes fonctionnant en temps-réel [LM01, VLLVR04, Oli05] utilisent les outils de synthèse d'images actuels pour reproduire des résultats convaincants : des filtres, des segmentations et des abstractions sont mis en place pour disperser les couleurs. L'un des problèmes engendré par ces méthodes est l'ef-



F . 2.10: (a) Aquarelle physiquement réaliste de Curtis *et al.* [CAS⁺97]. (b) Rendu temps-réel de Bousseau *et al.* [BKTS06].

fet “rideau de douche” obtenu lorsqu’une texture plaquée sur l’écran ne bouge pas de la même manière que la scène (vidéo ou 3D) ; cela provoque des effets de glissement indésirables qui illustrent la dissociation qui peut intervenir entre le mouvement du style et celui du message. On retrouve ce problème de cohérence temporelle d’une scène 3D glissant sous une texture 2D plaquée en espace image dans de nombreuses stylisations. C’est la raison pour laquelle plusieurs auteurs s’y sont penchés. Dans [BKTS06, BNTS07], Bousseau *et al.* utilisent l’advection de textures pour faire bouger le papier dans le même sens que celui donné par le flot de mouvement dans une vidéo. Les canevas dynamiques [CTP⁺03] représentent des métaphores dans lesquelles un mouvement 2D est associé à chacun des mouvements d’une scène 3D comme les rotations, les translations et les zooms. Ainsi la texture plaquée sur l’écran bouge de manière cohérente par rapport aux mouvements de la caméra. Bénard *et al.* [BBT09, BLV⁺10] améliorent cette technique en permettant de faire bouger des textures de manière cohérente indépendamment sur chacun des objets de la scène.

Les styles cartoon sont aussi très étudiés et déjà largement utilisés en industrie pour la production de films, de dessins animés et de jeux vidéo [MFE07]. De nombreux effets interviennent dans un rendu de type cartoon : des aplats et des discontinuités entre les couleurs, des abstractions de formes et de matériaux, des segmentations, etc. Parmi les travaux, on distingue notamment ceux qui travaillent directement sur les photographies ou les vidéos. Les premiers articles utilisent des filtres gaussiens ou médians pour faire en sorte d’obtenir des images simplifiées. L’algorithme de segmentation *mean shift* [CM99] est aussi très adapté pour trouver les régions pertinentes et les simplifier. De nombreux articles l’ont utilisé pour ce type de rendu [WTXC04, WXSC04, DS02]. Winnemöeller *et al.*



Fig. 2.11: Rendu de type cartoon sur des images fixes de Kang *et al.* [KLC09].

[WOG06] utilisent des applications successives d'un filtre bilatéral [TM98] pour faire une abstraction des couleurs sur des vidéos en temps-réel. Des différences de gaussiennes sont aussi utilisées pour détecter les contours. Le filtre bilatéral est appliqué le long du flot obtenu par les tangentes des arêtes elles-mêmes lissées dans [KLC09] pour obtenir un style cartoon cohérent (cf. Figure 2.11). Orzan *et al.* [OBBT07] travaillent dans le domaine du gradient pour lisser certains détails et obtenir une abstraction sur des photographies tout en conservant les structures principales. L'application de filtres a l'avantage de fournir une bonne cohérence temporelle lorsque les images sont animées. Différents filtres sont appliqués en temps-réel dans [KL08, KKD09, KKD10] dans le but d'obtenir un style cartoon cohérent. Lorsqu'il s'agit de travailler sur des objets 3D, un éclairage Lambertien quantifié en quelques paliers est généralement utilisé pour obtenir le bon effet [LMHB00]. Ces paliers sont souvent codés dans une texture 1D contenant les couleurs pour chacune des intensités lumineuses. Ce principe est appliqué dans [GGSC98, GSG⁺99] pour appliquer des couleurs chaudes et froides sur des objets 3D et obtenir des illustrations techniques. Todo *et al.* [TABI07] ont aussi mis en place une interface utilisateur pour modifier ces démarcations entre les différents tons. Beaucoup d'autres travaux se consacrent à la manipulation et la stylisation d'éléments plus spécifiques comme les reflets lumineux qui apparaissent sur les objets brillants [AH03, PGSP08], les ombres [DCFR07] ou même la fumée [MF06]. Les stylisations plus abstraites peuvent aussi être envisagées, comme les binarisations de scènes 3D [BRO08, XKM07] ou la création de clip-arts [SEH08, EWS08].

Les dessins techniques s'éloignent des rendus artistiques pour montrer des caractéristiques plus spécifiques généralement utiles pour certains types d'applications : la visualisation de cartes (Figure 1.2), les dessins explicatifs pour le montage d'objets comme les meubles, mais aussi la Conception Assistée par Ordinateur (CAO). Le message est donc plus d'ordre explicatif. Les diagrammes explosés montrent les relations entre les parties d'objets complexes et la manière dont ils s'imbriquent entre eux [BG06, LRA⁺07, TKS10]. Li *et al.* [LAS04] utilisent une approche originale en se basant sur une série d'images, représentant chacune des parties des objets en entrée. Ils s'appuient aussi sur les conventions traditionnelles utilisées par les illustrateurs pour montrer les différentes régions du corps humain ou les parties d'objets complexes avec notamment un style cartoon [LRA⁺07, LACS08]. Knödel *et al.* [KHG09] appliquent des styles originaux comme le *blueprint*, la transparence ou encore les contours pour montrer les différentes parties d'un objet complexe. D'autres auteurs se sont penchés sur le placement de labels dans une scène 3D, de sorte que les notes soient toujours lisibles, sans obstruer la visualisation des objets dont l'environnement est composé [SD08]. Le mouvement des objets peut aussi être analysé pour montrer précisément comment un système fonctionne [MY⁺10].

Les analogies sont employées pour reproduire des styles existants basés sur des exemples. L'objectif consiste ici en une simple reproduction de style défini. Il est par exemple possible de mettre en place des méthodes permettant de transférer les couleurs d'une image vers une autre. C'est ce qui a été fait dans [HJO⁺01, PR10] pour modifier les couleurs et le ton d'une image existante en partant d'une autre palette de couleurs. Mais le transfert de style peut s'appliquer à n'importe quelles caractéristiques. Par exemple, d'autres auteurs ont travaillé à reproduire des styles comme les peintures [DCoY03, PR10], les hachures [JEGPO02], les lignes [FTP03] et les motifs [BBT⁺06, HLT⁺09].

D'autres styles peuvent être exprimés à l'aide de moyens informatiques. Comme souligné précédemment, le domaine du *rendu expressif* est très vaste. De nombreux messages peuvent être communiqués en utilisant les styles qui leur correspondent. Les reproductions d'outils artistiques comme les crayons wax [RMN05], les modèles de peinture à l'huile [BWL04, BL04, XTJ⁺08, CBWG10, BG10], de couteaux [OJIN08] ou même de *jet painting* [LOG06] permettent de simuler des styles sans avoir à se préoccuper du message, si ce n'est celui imposé par les contraintes de l'outil lui-même. Les perspectives multiples ou non linéaires [DSLL98, CS04, YM04, SGS08, BSSS07, YMS08, MDWK08] peuvent aussi aider à percevoir plus d'informations sur des objets, des terrains ou des paysages. Des styles originaux sont aussi parfois étudiés. On peut notamment citer l'imitation de style cubique et segmenté [CH03, KSFC02], de mosaïques [SLK05, OK08], la reproduction à l'aide de caractères ASCII [XZW10], des rendus qui imitent les effets de papier plié sur des objets 3D [XCDR10] ou des effets d'animation pour mettre en avant des effets de mouvements dans des scènes dynamiques [BZOP07, SSBG10].

2.3 Conclusion

Au final, le choix d'un message et d'un style est donc infini. Dans tous ces exemples, le style employé est en relation avec le message qui doit être communiqué, et inversement. Des styles abstraits utilisant des aplats de couleurs, des lissages ou des segmentations et des marques sont généralement utilisés pour reproduire des dessins artistiques et donner des impressions visuelles plaisantes. Ce sont d'autres types d'abstractions et des styles plus épurés qui vont plutôt se retrouver dans les illustrations scientifiques. C'est une partie de ces relations étroites entre messages et styles que nous allons explorer dans cette thèse. Nous étudions plus spécifiquement deux domaines du *rendu expressif* dans le but de communiquer la forme d'objets 3D dans des scènes dynamiques : la mise en valeur des détails au travers de l'éclairage et le dessin au trait.

Première partie

Mise en valeur de la forme à travers l'éclairage

L'éclairage et la forme

Quelle que soit la manière dont on éclaire une scène 3D, des effets d'ombrage et des variations de couleurs dus aux matériaux des objets atténuent ou exagèrent certains détails dans la scène. Les interactions complexes qui existent entre la forme, la lumière et le matériau modifient la perception que nous nous faisons de la forme. Que ce soit dans les applications ludiques ou scientifiques, il peut être important de montrer les détails de surface des objets. Lorsqu'un archéologue doit photographier un objet, il place généralement une source lumineuse rasante pour faire en sorte de montrer le plus de détails possibles. Cependant, certains éléments peuvent toujours être masqués ou exagérés. Dans un jeu vidéo, la mise en valeur des détails peut aussi être mise en œuvre pour donner un style original ou pour assurer une bonne compréhension des éléments constitutifs de la scène 3D. La description de la forme à travers l'éclairage est un domaine en pleine expansion qui peut donc servir dans de nombreux secteurs.

Dans cette partie, nous verrons comment exagérer la forme en modifiant directement l'équation de réflexion. De cette manière, il est possible de montrer des détails quel que soit le style utilisé. Après avoir exploré les travaux existants dans ce domaine, nous détaillerons une nouvelle manière d'extraire les caractéristiques de surface sur des scènes 3D animées, puis nous montrerons des approches pour utiliser cette information afin d'exagérer les détails via trois différentes classes de style.

3.1 Réintroduction de la forme par les artistes

Les dessinateurs et les photographes ont l'habitude de manipuler la lumière pour mettre en valeur des éléments d'une scène. Dans les dessins ou les peintures, ce sont les interactions complexes entre matière, lumière et forme qui sont directement dessinées par les artistes. En regardant le travail des illustrateurs scientifiques [Woo94, Hod03] ou des artistes comme Norman Rockwell ou Burne Hogarth [Hog91], il est facile de s'imaginer les compétences et la patience nécessaire

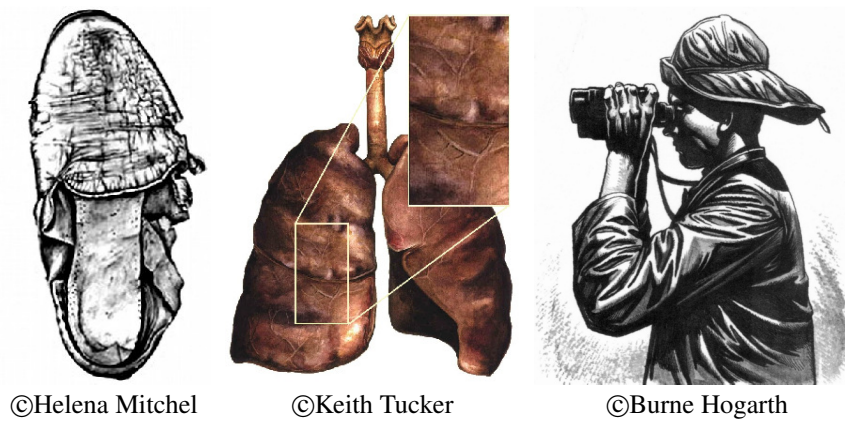


FIG. 3.1: Trois dessins montrant comment les artistes mettent les détails de surface en valeur. À gauche : une chaussure médiévale dans laquelle les petits détails de surface ont été mis en valeur. Au milieu : des poumons sur lesquels les veines ont été exagérées. À droite : un personnage portant un vêtement brillant et composé de nombreuses pliures.

pour réaliser ces œuvres. Leur objectif n'est pas seulement d'imiter précisément une scène existante, mais aussi de communiquer d'autres caractéristiques, et notamment la forme, pour rendre le dessin plus compréhensible. Les trois exemples de la Figure 3.1, démontrent la précision utilisée par les artistes pour communiquer les détails de forme dans un style presque réaliste. Des coups de crayon très fins sont utilisés par Helena Mitchell pour représenter non seulement les détails de surface sur cette chaussure, mais aussi un éclaircissement très subtil avec de petites variations d'intensités de gris. Les poumons, dessinés par Keith Tucker (au milieu), décrivent précisément la forme globale de ces organes ainsi que des veines très fines avec une grande précision. Des exemples similaires peuvent aussi être trouvés dans les illustrations artistiques, comme le dessin de Burne Hogarth dans lequel les plis du manteau, ainsi que sa réflectance, sont très clairement mis en valeur.

Les dessinateurs possèdent donc une expertise sur la manière de mettre en valeur certaines caractéristiques d'une scène. Le procédé de création d'une photographie diffère de celui des dessins. Les artistes disposent, d'une part, d'objets composés de leurs propres matériaux, et doivent, d'autre part, manipuler les lumières pour montrer au mieux certaines caractéristiques comme la forme ou la réflectance. Sur l'exemple du médaillon (Figure 3.2 (a)), la différence des formes perçues entre les deux photographies est flagrante. À gauche, les deux sources lumineuses sont symétriques et créent un effet uniforme sur l'ensemble de l'objet. Lorsque l'une des deux sources est placée de telle sorte que la direction de la lumière soit rasante, tous les détails sont révélés (à droite). Cette source lumineuse contribue à montrer

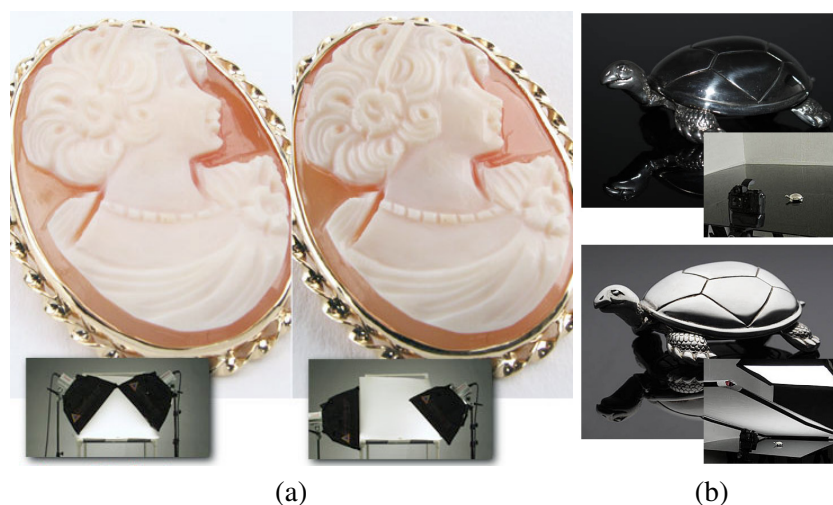


Fig. 3.2: La manipulation de la lumière joue un rôle essentiel pour les photographes. À gauche : exagération des détails sur un médaillon à l'aide d'une lumière rasante. À droite : photographie d'un objet très réfléchissant. ©Photoflex. <http://www.photoflexlightingschool.com/>.

la texture de l'objet tandis que l'autre crée une source ambiante, toujours nécessaire pour obtenir un éclairage global cohérent. Le deuxième exemple montre comment l'artiste manipule la lumière pour photographier un objet très réfléchissant (Figure 3.2 (b)). Pour mettre en valeur les concavités de la carapace de cette tortue en argent, un diffuseur est mis en place juste au-dessus pour obtenir une lumière uniforme couvrant toute la statuette. Un filtre est aussi placé entre le diffuseur et la tortue pour que la transition du blanc au noir soit plus douce et révèle plus de détails. Par comparaison avec la photographie du haut dans laquelle seul le flash a été utilisé, le résultat est beaucoup plus riche en terme de description de la forme.

Le problème se retrouve en informatique graphique. Dans les films d'animation, les artistes manipulent séparément les données d'une scène. L'image tirée du film d'animation "Renaissance" dans la Figure 3.3 illustre cette contrainte. Dans un style aussi drastique que le noir et le blanc, les artistes se doivent de manipuler les lumières de sorte que le spectateur perçoive les formes précisément tout en reflétant l'ambiance du film. En y regardant de plus près, on s'aperçoit que l'oreille de l'héroïne ne devrait pas être éclairée comme cela si on se réfère à la direction de lumière principale. Les artistes ont alors dû ajouter et modifier des sources lumineuses à la main dans l'environnement pour en montrer tous les éléments.



FIG. 3.3: Image tirée du film d'animation "Renaissance" [Vol06].

Que cela soit pour des dessins artistiques, des films d'animation ou des photographies, la manipulation de la lumière est primordiale pour souligner la forme et les détails des objets d'une scène. Ce sont les positions précises des sources lumineuses qui vont générer des variations d'intensités subtiles et révéler au mieux la forme. Ces exemples nous apprennent aussi que ce sont souvent des environnements "irréalistes" (i.e. aux statistiques non-naturelles [Fie87, FDA03]) qui permettent de montrer les détails de manière efficace.

3.2 Équation de réflexion

Dans une scène 3D, nous disposons de données géométriques en entrée comme le maillage, l'environnement et les matériaux. C'est une interaction complexe entre ces trois éléments qui permet de générer le rendu final. Avant de montrer comment manipuler les données pour mettre en valeur la forme des objets, il est indispensable de comprendre comment tous ces éléments agissent les uns avec les autres.

3.2.1 Présentation de la fonction

L'équation de rendu [Kaj86] est utilisée explicitement ou implicitement dans la plupart des outils de synthèse d'images. Elle représente la loi de la conservation d'énergie qui stipule que la lumière qui émane d'un point de la surface d'un objet est égale à la somme des énergies émises et reflétées :

$$L(\mathbf{p} \rightarrow \mathbf{e}) = L_e(\mathbf{p} \rightarrow \mathbf{e}) + L_r(\mathbf{p} \rightarrow \mathbf{e})$$

où $L_e(\mathbf{p} \rightarrow \mathbf{e})$ et $L_r(\mathbf{p} \rightarrow \mathbf{e})$ sont respectivement la radiance émise et réfléchie par la surface au point \mathbf{p} dans la direction du point de vue \mathbf{e} . La radiance représente la quantité d'énergie par unité de surface et de direction qui arrive ou quitte une surface : elle est exprimée en watt par mètre carré par stéradian ($Wm^{-2}sr^{-1}$). Le lecteur peut se référer à [RWPD05] pour obtenir des informations supplémentaires

sur les notions fondamentales de radiométrie. Par la suite, seules les surfaces qui n'émettent pas d'énergie (i.e. $L_e = 0$) seront généralement considérées. Dans ce cas, seuls les phénomènes de réflexion interviennent et l'équation devient :

$$L(\mathbf{p} \rightarrow \mathbf{e}) = L_r(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_{\mathbf{n}}} \rho(\mathbf{e}, \boldsymbol{\ell})(\mathbf{n} \cdot \boldsymbol{\ell}) L(\mathbf{p} \leftarrow \boldsymbol{\ell}) d\boldsymbol{\ell} \quad (3.1)$$

avec :

1. L'éclairage incident $L(\mathbf{p} \leftarrow \boldsymbol{\ell})$ qui représente la radiance entrante partant de la source lumineuse $\boldsymbol{\ell}$ et arrivant sur le point \mathbf{p} de la surface.
2. L'orientation de la surface définie comme étant le produit scalaire entre la normale \mathbf{n} et la direction de la lumière incidente $\boldsymbol{\ell}$.
3. Les caractéristiques du matériau $\rho(\mathbf{e}, \boldsymbol{\ell})$, paramétré par le vecteur vue \mathbf{e} et la direction incidente de la lumière $\boldsymbol{\ell}$. Cette fonction est appelée la *Bidirectional Reflectance Distribution Function (BRDF)*.

Le produit de ces trois éléments est intégré sur l'hémisphère¹ $\Omega_{\mathbf{n}}$ positionné au point \mathbf{p} et orienté selon \mathbf{n} pour obtenir la radiance $L(\mathbf{p} \rightarrow \mathbf{e})$ réfléchie depuis le point \mathbf{p} dans la direction du point de vue \mathbf{e} .

La plupart des images de synthèse sont générées à l'aide de cette équation ou d'une variante. Les articles traitant de la mise en valeur de détails dans le rendu ou les travaux que nous allons présenter dans cette thèse peuvent donc s'inscrire à différents endroits de cette équation.

3.2.2 BRDF

La *BRDF* [Nic70] est, dans le cas général, une fonction 4D décrivant les caractéristiques de réflexion d'un matériau. Elle modélise les interactions entre lumière et matière à l'échelle microscopique et détermine la quantité d'énergie lumineuse dans une direction d'observation pour une direction incidente de lumière donnée. La *BRDF* combine de nombreux effets dont les matériaux sont composés, comme la brillance, l'effet de Fresnel ou encore l'anisotropie. Dans le cas général, elle prend les directions de vue et de lumière incidente en paramètre et est décomposée comme la somme de *BRDFs* caractéristiques de certains phénomènes lumineux. En infographie, elle est ainsi souvent exprimée sous la forme :

$$\rho(\mathbf{e}, \boldsymbol{\ell}) = \rho_d(\mathbf{e}, \boldsymbol{\ell}) + \rho_s(\mathbf{e}, \boldsymbol{\ell}) \quad (3.2)$$

où ρ_d et ρ_s correspondent respectivement à la composante diffuse et à la composante spéculaire de la *BRDF*. Comme on peut le voir sur la Figure 3.4, ces deux composantes représentent respectivement les comportements indépendants et dépendants du point de vue et permettent de définir la manière dont l'énergie lumineuse se comporte à la surface de l'objet. Lorsque l'énergie renvoyée est la

¹Sur la sphère complète lorsque l'on souhaite décrire des phénomènes réfractifs.

même dans toutes les directions, on parle de réflexion Lambertienne ou diffuse. Dans ce cas, la valeur renvoyée par la *BRDF* est une constante positive dépendante de la surface. A l'opposé, l'énergie est renvoyée dans une unique direction pour une surface spéculaire pure, comme un miroir. La direction dans laquelle l'énergie se reflète est la direction de réflexion, définie comme étant le symétrique du rayon lumineux par rapport à la normale :

$$\mathbf{r} = -\ell + 2(\ell \cdot \mathbf{n})\mathbf{n}$$

Les *BRDFs* brillantes (ou *glossy*) généralisent les effets extrêmes obtenus avec des matériaux parfaitement diffus et spéculaires. Elles réfléchissent la lumière autour d'une direction privilégiée (souvent la direction miroir).

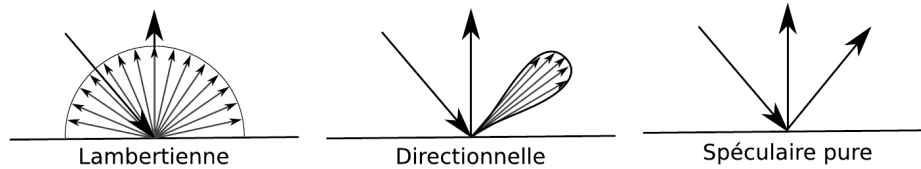


Fig. 3.4: À gauche : la réflexion Lambertienne renvoie la lumière de la même manière dans toutes les directions. À droite, une surface spéculaire pure réfléchit la totalité de l'énergie lumineuse dans la direction du vecteur de réflexion. Au centre : une surface brillante renvoie l'énergie de manière privilégiée autour du vecteur de réflexion.

Les quatre dimensions de la *BRDF* ne sont pas forcément toutes utiles si on décide de restreindre les caractéristiques des matériaux pouvant être créés. Si la fonction est isotrope par exemple, l'énergie renvoyée est invariante par rapport à la rotation autour de la normale et le nombre de dimensions diminue à trois. D'un autre côté, la *BRDF* ne permet pas de modéliser tous les effets qui entrent en jeu lors de l'interaction entre la surface et la lumière. Par exemple, si l'on veut que la fonction varie sur l'objet pour créer des effets non uniformes, la fonction doit aussi prendre en compte la position du point sur la surface (deux dimensions supplémentaires). On parle alors de *Spatially-Varying Bidirectional Reflectance Distribution Function (SVBRDF)* [LBAD⁺06]. On peut aussi vouloir modéliser des effets translucides que l'on rencontre dans des surfaces comme la peau ou le marbre. Dans ce cas, des phénomènes d'absorption interviennent et l'énergie reçue en un point peut être transmise sur un autre point de la surface. La fonction prend donc en compte deux positions supplémentaires (huit dimensions) et est appelée *Bidirectional Sub-surface Scattering Reflectance Distribution Function (BSSRDF)* [JMLH01]. Des effets encore plus complexes comme des ombrages ou des inter-réflexions dues à la méso-structure sous-jacente peuvent être aussi être intégrés dans une série de textures : les *Bidirectional Texture Functions (BTFs)* [DvGNK99].

Si la définition première de la *BRDF* est basée sur des contraintes physiquement réalistes comme la réciprocité d’Helmholtz ou la conservation de l’énergie [Nic70], il est coutume d’appeler ainsi des modèles phénoménologiques pouvant s’abstraire de ces contraintes. Le plus connu d’entre eux est sûrement le modèle analytique et empirique de Phong [BT75] (et son extension [Bli77]), généralement utilisé dans les applications interactives. Il a aussi mis en place un modèle d’ombrage souvent utilisé pour approcher l’équation de réflexion et généralisant l’équation 3.2. Avec un environnement composé d’une seule lumière directionnelle $\boldsymbol{\ell}_0$, l’équation 3.1 devient :

$$L(\mathbf{p} \rightarrow \mathbf{e}) = \sum_j \rho_j(\mathbf{e}, \boldsymbol{\ell}_0) L_j(\boldsymbol{\ell}_0) \quad (3.3)$$

où $j \in \{a, d, s\}$ itèrent sur les composants ambiants, diffus et spéculaires du modèle d’ombrage. L_j représente l’intensité lumineuse par composante (L_a étant constante), représenté la plupart du temps par une couleur RVB (pour Rouge, Vert et Bleu). Les termes ambiants, diffus et spéculaires sont alors donnés par $\rho_a = cst$, $\rho_d(\boldsymbol{\ell}_0) = \rho_d \times (\mathbf{n} \cdot \boldsymbol{\ell}_0)$ et $\rho_s(\mathbf{e}, \boldsymbol{\ell}_0) = \rho_d \times (\mathbf{r}_0 \cdot \mathbf{e})^\eta$. Dans cette formulation, le terme ambiant permet de simuler de manière très grossière les multiples réflexions qui surviennent dans un environnement et η permet de contrôler la brillance (ou *highlight*) du matériau.

De nombreux travaux ont été réalisés pour trouver des modèles de *BRDF* efficaces qui représentent et contrôlent facilement les caractéristiques des matériaux [Sch94a, SMSA01, DR08]. Dans les sections et les chapitres qui suivent, le modèle de Phong, le modèle Lambertien (i.e. ρ_d) ou des modèles proches seront largement utilisés. La raison principale est que, de manière générale, les rendus artistiques ne nécessitent pas de *BRDF* réalistes mais plutôt des modèles facilement contrôlables. Nous serons néanmoins aussi amenés à créer des images plus réalistes. Le modèle d’Ashikhmin et Shirley [AS00] possède, par exemple, plusieurs avantages comme la réciprocité et la conservation de l’énergie. Cette *BRDF* est constituée de paramètres intuitifs et est classiquement composée d’une somme de termes diffus et spéculaires (cf. Equation 3.2). Si on ne considère que les matériaux isotropes, les composantes sont données par :

$$\rho_d(\mathbf{e}, \boldsymbol{\ell}) = \frac{28R_d}{23\pi} (1 - R_s) \left(1 - \left(1 - \frac{(\mathbf{n} \cdot \boldsymbol{\ell})}{2} \right)^5 \right) \left(1 - \left(1 - \frac{(\mathbf{n} \cdot \mathbf{e})}{2} \right)^5 \right)$$

$$\rho_s(\mathbf{e}, \boldsymbol{\ell}) = \frac{\sqrt{\eta + 1}}{8\pi} \frac{(\mathbf{n} \cdot \mathbf{h})^\eta}{(\mathbf{h} \cdot \mathbf{e}) \max((\mathbf{n} \cdot \boldsymbol{\ell}), (\mathbf{n} \cdot \mathbf{e}))} F((\mathbf{e} \cdot \mathbf{h}))$$

où $\mathbf{h} = (\boldsymbol{\ell} + \mathbf{e}) / \|\boldsymbol{\ell} + \mathbf{e}\|$ est le demi-vecteur entre la direction $\boldsymbol{\ell}$ et la vue \mathbf{e} ; R_d et R_s sont des couleurs qui spécifient les réflectances diffuses et spéculaires, et η est un paramètre de brillance, comme dans le modèle de Phong. La fonction F contrôle l’effet de Fresnel qui accroît les spécularités lorsque l’angle incident diminue. Elle est facilement obtenue avec l’approximation de Schlick [Sch94c] :

$$F((\mathbf{e} \cdot \mathbf{h})) = R_s + (1 - R_s)(1 - (\mathbf{e} \cdot \mathbf{h}))^5$$

3.3 Les rendus stylisés existants

La plupart des travaux modifient l'équation de réflexion 3.1 dans le but de communiquer la forme des objets d'une scène. Ils se basent souvent sur des styles simples et spécifiques pour créer des images épurées ne montrant que l'essentiel.

3.3.1 Styles *cartoon*

Les styles *cartoon* sont largement utilisés pour mettre les caractéristiques de forme en valeur dans les scènes 3D. Le principe est d'indexer une palette de couleurs (généralement une texture 1D), avec l'intensité d'un éclairage résultant souvent d'un modèle de Phong ou Lambertien. Ainsi, il est possible d'obtenir des rendus variés et originaux dont certains peuvent avoir des transitions de couleurs continues ou discontinues sur les isolignes du modèle diffus. C'est ce qu'ont fait Gooch *et al.* [GGSC98, GSG⁺99] dans le but d'imiter les illustrations techniques. Des couleurs froides, comme le bleu, sont utilisées dans les zones sombres tandis que des couleurs chaudes (le jaune, le rouge) sont choisies pour les régions très éclairées.

Barla *et al.* [BTM06] étendent la technique en ajoutant une seconde dimension à la texture (Figure 3.5). L'axe des abscisses correspond toujours à la couleur utilisée pour l'intensité lumineuse et la seconde dimension permet de représenter une autre caractéristique comme la profondeur ou l'orientation de la surface. Une texture 2D est aussi utilisée dans [SMGG01] pour encoder l'éclairage d'une scène 3D. Les auteurs introduisent la *Lit Sphere* : une texture sur laquelle apparaît une

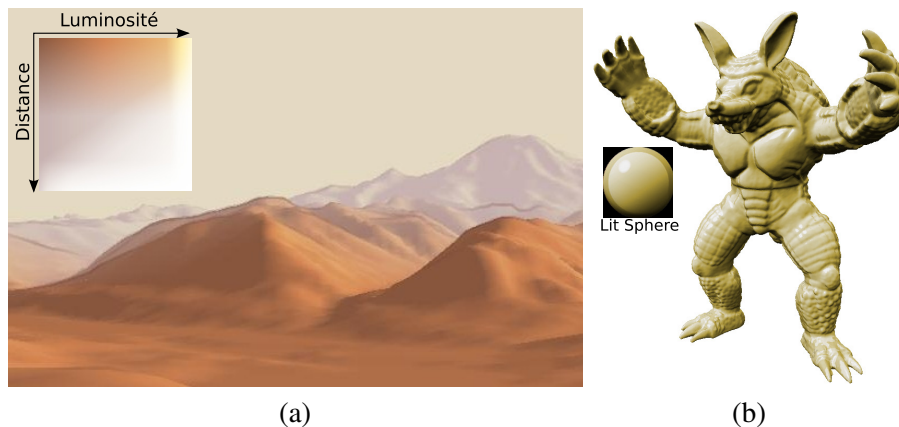


FIG. 3.5: À gauche : *X-Toon* de Barla *et al.* [BTM06] utilise une texture 2D indexée à la fois par l'intensité lumineuse et par la distance sur un terrain. À droite : Les *Lit Spheres* [SMGG01] permettent d'indexer les valeurs d'une image à partir des normales projetées dans l'espace de la caméra.

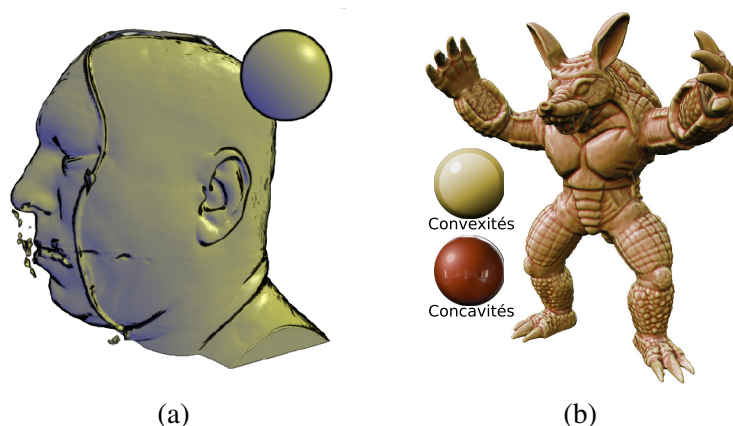


Fig. 3.6: (a) *Mean curvature shading* appliqué aux *Lit Spheres* [BG07]. (b) Un exemple approchant les effets que l'on peut obtenir dans le logiciel ©ZBrush. Les concavités et convexités apparaissent clairement avec l'utilisation de deux *Lit Spheres*.

simple sphère dont le rendu est déjà calculé ou dessiné par un utilisateur. Ce sont les normales, projetées dans l'espace de la caméra, qui vont permettre d'indexer les couleurs contenues dans cette image (Figure 3.5). Cette fois ci, ce n'est donc plus un modèle d'ombrage qui indexe la texture, mais les normales. Il est donc possible de construire des textures très diversifiées, mais cette méthode est limitée à un environnement spécifique : impossible de faire tourner les lumières autour d'un objet. De plus, aucune de ces méthodes ne propose de communiquer la forme explicitement. L'utilisateur n'a pas de contrôle direct sur les éléments qui sont mis en avant lors du rendu.

L'une des informations qui caractérise précisément la forme est assurément la courbure de la surface. Elle permet de différencier creux et bosses, surfaces planes et surfaces courbes. La courbure correspond à une information de second ordre de la surface (cf. Chapitre 4) et est utilisée dans de nombreuses méthodes pour communiquer les détails de forme. Dans [KWTM03], cette information est prise en compte pour moduler le modèle d'éclairage de Phong et modifier les couleurs initiales en fonction du type de surface considéré (convexe ou concave). Ce procédé, plus communément appelé *mean curvature shading*, revient à utiliser la courbure comme axe des ordonnées dans la technique *X-toon* [BTM06]. Les *Lit Spheres* sont aussi employées dans plusieurs méthodes prenant la courbure en compte. Par exemple, les normales situées dans des régions très courbées sont modifiées dans [BG07] de telle sorte qu'elles soient projetées sur les bords de la *Lit Sphere*, et donc proche des silhouettes (Figure 3.6 (a)). Le logiciel de modélisation ©ZBrush inclut aussi le procédé des *Lit Spheres* sous le nom de *MatCap*. Elles sont combinées avec une information de courbure pour obtenir des effets qui exagèrent les

détails des objets. Deux *Lit Spheres* sont combinées : la première est utilisée dans les convexités tandis que la seconde l'est dans les concavités. En choisissant une fonction de transition douce entre ces régions, les résultats communiquent la forme de manière convaincante (cf. Figure 3.6 (b)). Le style est néanmoins toujours limité à un seul type de rendu (cartoon ou Lambertien), obtenu avec un seul point de vue ou un nombre limité de sources lumineuses.

3.3.2 Accessibility Shading et Ambient Occlusion

Une autre manière de mettre en valeur les détails de surface des objets est de se rapporter aux techniques d'*Accessibility Shading* [Mil94]. La méthode originale se base sur un descripteur géométrique de la scène : une sphère est mise en contact avec chacun des points de la surface, et sa taille est ajustée de sorte qu'elle ne touche aucune autre partie de l'objet. Si un point se trouve dans une petite cavité, le diamètre de la sphère qui sera calculé sera très faible. L'effet inverse se produit dans les zones planes ou convexes. L'"accessibilité" est donc représentée par une valeur scalaire déterminant à quel degré un point de la surface se trouve dans un creux, et si celui-ci est profond ou non.

Le même problème est traité par la technique répandue de l'*Ambient Occlusion* (AO) [PG04]. Elle est définie comme la visibilité moyenne du point par rapport à son environnement (cf. Figure 3.7 (a)) et donne une information scalaire sur l'occultation du point \mathbf{p} . Si $V(\ell) \in \{0, 1\}$ est un terme de visibilité vérifiant si le rayon lumineux ℓ arrive ou non sur le point considéré, alors l'AO est donné par :

$$A(\mathbf{p}) = \frac{1}{\pi} \int_{\Omega_n} (\mathbf{n} \cdot \ell) V(\ell) d\ell \quad (3.4)$$

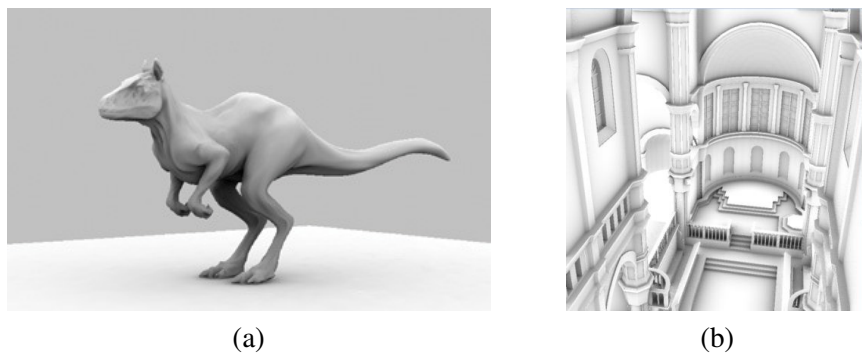


FIG. 3.7: (a) *Ambient Occlusion* [PG04]. L'AO permet d'obtenir des effets réalistes avec l'utilisation de matériaux diffus, où les cavités sont généralement assombries. (b) *Screen Space Ambient Occlusion*. L'information d'occlusion montre clairement le volume et les relations entre les différents éléments de la cathédrale.

Ces méthodes sont assez complexes à mettre en œuvre car un nombre élevé d'échantillons doit être pris en compte sur chacun des sommets de l'objet 3D pour estimer la visibilité moyenne. Elles peuvent néanmoins être précalculées pour ensuite être utilisées dans des applications interactives comme les jeux vidéo. Des approximations de l'AO peuvent aussi être calculées avec des techniques fonctionnant en temps-réel [SA07, BSD08]. La carte de profondeurs centrée sur le point de vue (appelée tampon de profondeurs) est analysée pour estimer l'AO avec quelques échantillons (cf. Figure 3.7 (b)). Ces méthodes ont l'avantage d'être indépendantes de la complexité de la scène, de ne pas nécessiter de précalculs qui demandent des ressources mémoires importantes, et surtout, de pouvoir faire fonctionner l'algorithme sur des scènes dynamiques en temps-réel.

Ces techniques ont néanmoins plusieurs inconvénients : tout d'abord, la mise en valeur des détails est appliquée de manière implicite. L'utilisateur n'a donc pas ou très peu de contrôle sur les éléments qui sont exagérés. Ensuite, ces méthodes permettent seulement de montrer les cavités de la surface. Certains détails importants peuvent alors être ignorés lors du rendu. Enfin, l'information scalaire qui est extraite est assez limitée car elle est adaptée seulement aux matériaux diffus : cela a pour conséquence de réduire l'intervalle des styles possibles.

3.3.3 Exagération des détails

Des chercheurs se sont intéressés à mettre explicitement la forme en valeur avec différents styles. Les *Polynomial Texture Maps (PTM)*, introduites par Malzbender *et al.* [MGW01], sont des textures qui stockent les coefficients d'une quadrique représentant l'intensité reflétée pour toutes les directions lumineuses depuis un point de vue donné. En modifiant la courbure de cette quadrique, les variations d'intensités lumineuses entre les zones claires et les zones sombres sont amplifiées. Une conséquence directe est la perception accrue de la forme dans les textures. Les auteurs présentent aussi une autre manière de mettre en valeur les détails en extrapolant la quadrique sur les lumières rasantes. Celles-ci permettent de créer de nombreux contrastes et exagèrent tous les détails de surface.

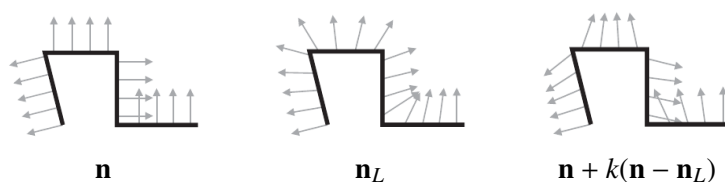
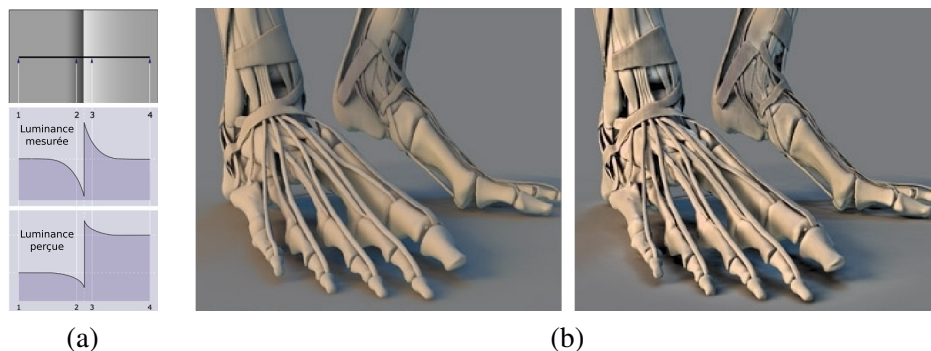


Fig. 3.8: Approche utilisée dans [CST05] pour mettre en valeur les hautes fréquences sur les normales. Les normales du maillage original (à gauche) sont combinées avec des normales lissées (au milieu) pour obtenir un résultat dans lequel les arêtes vives sont exagérées (à droite), sans modifier les silhouettes.

Cignoni *et al.* [CST05] modifient les normales situées à la surface des maillages pour augmenter le contraste intervenant sur les arêtes vives dans un rendu Lambertien. Leur méthode correspond exactement à celles employées pour ajouter des hautes fréquences sur les objets 3D : les différences entre le maillage et une version lissée de celui-ci (les hautes fréquences) sont ajoutées à l'objet original. La principale contribution est d'utiliser cette méthode sur les normales plutôt que sur les positions des sommets pour éviter de déformer les silhouettes lors du rendu. La Figure 3.8 illustre le procédé : la formule simple $\mathbf{n} + k(\mathbf{n} - \mathbf{n}_L)$ est utilisée pour exagérer les arêtes vives des objets 3D sans modifier les silhouettes. \mathbf{n}_L représente la normale lissée et k est un paramètre utilisateur permettant de contrôler l'intensité de la mise en valeur. Cette méthode est néanmoins limitée car elle ne fonctionne qu'avec un style : le modèle Lambertien. De plus, elle modifie la perception de la forme des objets de manière drastique, malgré le fait que les silhouettes restent inchangées. L'effet obtenu donne parfois l'impression que les régions planes sont courbées.

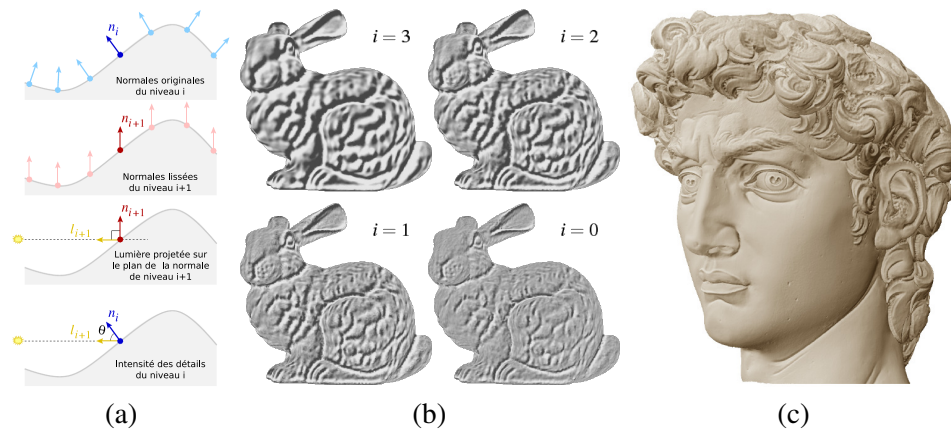
Les techniques de Luft *et al.* [LCD06] ou de Ritshel *et al.* [RSI⁺08] permettent de lever ces limitations car elles sont appliquées sur des valeurs scalaires (comme la profondeur ou la radiance) plutôt que sur des normales. Elles peuvent être appliquées sur des scènes réalistes composées de matériaux quelconques. C'est l'illusion de *Cornsweet* [Cor70] qui est utilisée sur le tampon de profondeurs dans [LCD06] ou sur la radiance sortante (i.e. $L(\mathbf{p} \rightarrow \mathbf{e})$) dans [RSI⁺08] pour augmenter la perception des contrastes dans une scène 3D. Cette illusion montre que les variations d'intensités lumineuses dans une image ne sont pas forcément perçues de la même manière par notre système de vision. Lorsqu'elle est appliquée sur une scène, cette fonction permet de mettre en valeur non seulement les détails de sur-



F . 3.9: (a) Illustration de l'illusion de *Cornsweet* [Cor70]. (b) Cette illusion est appliquée à la surface d'une scène 3D dans [RSI⁺08] pour mettre en valeur de nombreuses caractéristiques comme la forme et les ombres dans cet exemple.

face, mais aussi d'autres caractéristiques comme les ombres et les matériaux (cf. Figure 3.9). La forme est donc beaucoup mieux perçue après application de ce filtre. Il n'est toutefois pas possible de contrôler précisément les caractéristiques mises en valeur. La perception des matériaux est aussi altérée : les objets semblent plus brillants après application de la méthode.

Rusinkiewicz *et al.* [RBD06] présentent une méthode originale pour exagérer les détails de forme en modifiant directement les sources de lumières directionnelles en chaque point de l'objet 3D avec une approche multi-échelles. Comme on peut le voir à gauche de la Figure 3.10, la direction de la lumière est projetée sur le plan défini par la normale lissée du niveau suivant. L'intensité lumineuse des détails correspondant à cette échelle est alors obtenue en examinant l'angle entre la lumière projetée et la normale initiale. L'image du centre montre ainsi plusieurs niveaux de détail. Ces détails sont enfin utilisés pour obtenir le rendu final (à droite). Le style utilisé est le *Cosine Shading* qui consiste à transformer linéairement les valeurs du cosinus (du modèle Lambertien) dans l'intervalle $[0, 1]$. Ainsi, aucune zone d'ombre n'apparaît dans l'image créée. Le fait de créer localement une lumière rasante en chacun des points de l'objet à plusieurs échelles permet d'exagérer tous les détails. Néanmoins, cela provoque aussi des effets indésirables ; la méthode n'est pas cohérente lorsque la lumière bouge dans l'espace. Cela donne l'impression que la géométrie de l'objet est modifiée et s'aplatit.



F . 3.10: Méthode d'exagération des détails de Rusinkiewicz *et al.* [RBD06]. (a) Le calcul des détails est obtenu en examinant l'angle entre la normale et la direction de lumière projetée sur le plan de la normale lissée. (b) Plusieurs niveaux peuvent ainsi être calculés pour sélectionner des détails plus ou moins larges. (c) Lorsqu'ils sont utilisés dans le rendu, la forme apparaît clairement.

Toutes ces méthodes sont donc limitées dans le choix du style ou des caractéristiques qui sont mises en valeur. Dans [MGW01, LCD06, RSI⁺08], il n'y a pas de limitations sur le style car ces méthodes peuvent s'appliquer sur des rendus quelconques. Par contre, il est difficile de sélectionner et de contrôler la manière dont les détails sont mis en valeur. Dans [CST05, RBD06], le style est contraint au modèle Lambertien (ou à un équivalent) et la géométrie des objets semble affectée par les modifications des normales. Dans tous ces travaux, la perception des matériaux est plus ou moins modifiée.

Dans la suite de cette partie, nous introduisons une nouvelle manière de mettre en valeur les détails de surface sur les objets 3D. Le message à communiquer sera guidé par une information de courbure dépendante du point de vue, élément essentiel à la perception de la forme (cf. Chapitre 4). Cette information sera directement intégrée dans l'équation de réflexion (cf. Equation 3.1) de telle sorte qu'une grande variété de styles puisse être obtenue. Les différents styles imposent aussi des contraintes sur le message : la mise en valeur des détails doit se faire de manière continue le long de la surface et dans le temps ; le message ne doit donc pas contenir de discontinuités, spatiales ou temporelles. Les détails qui sont exagérés dans le rendu doivent aussi pouvoir être contrôlés par l'utilisateur, que ce soit pour l'échelle ou pour le type de surface sélectionné. Nous décrivons ces nouvelles méthodes dans le Chapitre 5.

Descripteurs de surface

La plupart des travaux précédents s'appuient sur le calcul d'une variation pour déterminer quels sont les éléments importants à mettre en avant dans le rendu. Il peut s'agir de la dérivée de la profondeur, de la radiance dans une image ou encore des hautes fréquences obtenues sur un maillage. De toutes ces dérivées, l'une est essentielle à notre perception de la forme : la courbure. C'est cette caractéristique que nous allons étudier dans ce chapitre.

L'information de courbure correspond à la dérivée seconde de la surface : elle peut donc être représentée par un tenseur 2×2 . Elle permet de décrire à la fois le type de la surface en chaque point d'un objet (concave, convexe, selle, etc.) et l'intensité de la courbure. C'est à partir de cette description de l'objet que le message va définir les détails à mettre en valeur.

Il existe de nombreuses méthodes pour calculer un tenseur de courbure. Il est souvent estimé à partir des positions des sommets d'un maillage et sa forme dépend du type des données analysées ; le tenseur est respectivement représenté par des matrices 1×1 , 2×2 et 3×3 pour les courbes, les surfaces et les volumes. Après avoir vu comment calculer le tenseur de courbure à la surface d'un objet 3D, nous verrons deux nouvelles méthodes ayant la particularité de prendre en compte le point de vue. Nous verrons que ces nouvelles données sont plus pertinentes pour le Système Visuel Humain (SVH) et qu'elles peuvent être exploitées très facilement par le message.

4.1 Descripteur centré sur l'objet

Revenons tout d'abord dans le cas 1D pour comprendre la notion de dérivée. La dérivée première de la courbe paramétrique en un point permet de déterminer la tangente de la fonction en ce point (cf. Figure 4.1 (a)). En regardant maintenant comment la tangente elle-même varie le long de la surface, il est possible de mesurer la courbure. Elle correspond donc à la dérivée seconde et détermine la

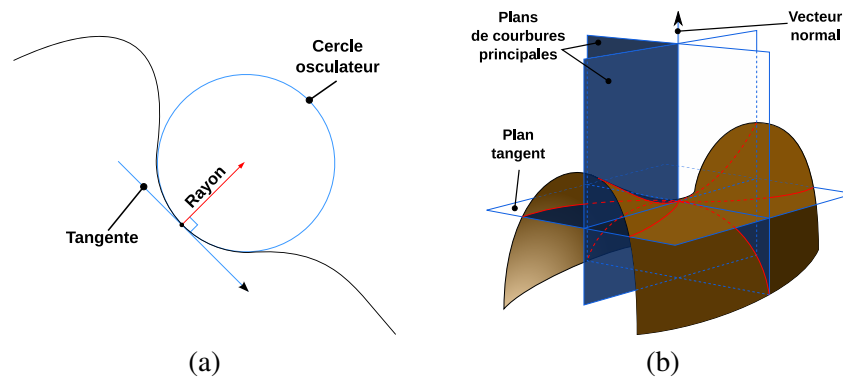


Fig. 4.1: (a) Dans le cas 1D, la courbure correspond à l'inverse du rayon du cercle osculateur en chacun des points de la courbe. (b) Sur une surface, la courbure est différente dans toutes les directions. On distingue notamment deux directions particulières (les plans de courbures principales) dans lesquelles la courbure est maximale et minimale. Illustrations tirées de <http://fr.wikipedia.org/>.

manière dont la courbe se plie dans l'espace. C'est équivalent à considérer l'inverse du rayon du cercle osculateur en un point : plus le rayon du cercle est petit, plus la courbure est grande.

Dans le cas d'une surface lisse, on peut calculer la courbure de la même manière en examinant les courbes 1D formées par l'intersection de plans (orientés selon la normale au point considéré) avec cette surface. Dans la Figure 4.1 (b), on remarque qu'il existe une infinité de courbures en un point donné, dans chacune des directions du plan tangent. Toutes ces informations peuvent être approchées facilement par un tenseur 2D symétrique sur des surfaces lisses [dC76, RDF05, RDF08]. Cette matrice est connue comme étant le tenseur fondamental de second ordre, noté \mathbf{II} . Son calcul s'effectue dans le plan tangent déterminé grâce au gradient de la surface¹. La normale est orthogonale à ce plan et représente déjà une information de premier ordre.

Considérons un repère local orthogonal $\{\mathbf{t}_u, \mathbf{t}_v\}$ perpendiculaire à la normale. Tout vecteur \mathbf{r} défini dans le plan tangent peut être exprimé dans la base $\{\mathbf{t}_u, \mathbf{t}_v\}$ sous la forme de coordonnées locales $\begin{pmatrix} u \\ v \end{pmatrix}^T$, où $\mathbf{r} = u\mathbf{t}_u + v\mathbf{t}_v$. La courbure dans la direction \mathbf{r} est une quadrique de la forme :

$$\kappa_{\mathbf{r}} = \begin{pmatrix} u & v \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u & v \end{pmatrix} \mathbf{II} \begin{pmatrix} u \\ v \end{pmatrix}$$

¹Le plan tangent correspond à une information de dérivée et peut être représenté de la même manière avec le tenseur fondamental de premier ordre.

Les trois valeurs scalaires de \mathbf{II} peuvent être calculées de différentes manières ; elles correspondent aux termes de second ordre obtenus lors du développement de la série de Taylor sur une surface. Lorsque la paramétrisation est conforme (i.e. que l'on paramétrise la surface localement dans le plan tangent), \mathbf{II} correspond à la *matrice hessienne* définie avec les dérivées secondes partielles de la surface. D'une manière générale, elle est définie en calculant la variation de la normale le long du plan tangent. La normale est en effet déjà une information de premier ordre ; sa dérivée correspond à la courbure [RDF08] :

$$\mathbf{II} = \begin{pmatrix} e & f \\ f & g \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{n}}{\partial t_u} \cdot \mathbf{t}_u & \frac{\partial \mathbf{n}}{\partial t_u} \cdot \mathbf{t}_v \\ \frac{\partial \mathbf{n}}{\partial t_v} \cdot \mathbf{t}_u & \frac{\partial \mathbf{n}}{\partial t_v} \cdot \mathbf{t}_v \end{pmatrix}$$

$\frac{\partial \mathbf{n}}{\partial t_i} \cdot \mathbf{t}_j$ étant la dérivée de la normale \mathbf{n} dans la direction \mathbf{t}_i projetée sur l'axe \mathbf{t}_j . À noter aussi que $\frac{\partial \mathbf{n}}{\partial t_u} \cdot \mathbf{t}_v = \frac{\partial \mathbf{n}}{\partial t_v} \cdot \mathbf{t}_u$. La matrice \mathbf{II} étant symétrique, il est toujours possible de faire une rotation sur le plan tangent pour la rendre diagonale :

$$\mathbf{II} = (\mathbf{t}_1 \quad \mathbf{t}_2) \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} (\mathbf{t}_1 \quad \mathbf{t}_2)^T$$

où κ_1 et κ_2 sont les valeurs propres issues de la matrice \mathbf{II} et correspondent aux courbures principales de la surface au point analysé. Les deux vecteurs propres associés \mathbf{t}_1 et \mathbf{t}_2 déterminent les directions dans lesquelles la variation des normales atteint un extremum local :

$$\kappa_{1,2} = \frac{e+g}{2} \pm \frac{\sqrt{4f^2+(e-g)^2}}{2} \quad \text{et} \quad \mathbf{t}_{1,2} = \begin{pmatrix} f \\ \kappa_{1,2} - e \end{pmatrix} = \begin{pmatrix} \kappa_{1,2} - g \\ f \end{pmatrix}$$

La Figure 4.1 (b) montre les plans de courbures principales obtenus sur une surface en forme de selle. En choisissant le repère orthogonal local $\{\mathbf{t}_u, \mathbf{t}_v\}$ comme étant égal à $\{\mathbf{t}_1, \mathbf{t}_2\}$, la courbure dans une direction $\mathbf{r} = u\mathbf{t}_1 + v\mathbf{t}_2$ du plan tangent est alors donnée par :

$$\kappa_{\mathbf{r}} = (u \quad v) \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

Le calcul de la courbure sur une surface lisse est donc théoriquement simple à mettre en œuvre. Il est par contre plus difficile à définir à la surface d'un maillage. Celle-ci est en effet composée de polygones (les triangles ou les quadrangles), sur lesquels la courbure est nulle, et d'arêtes sur lesquelles la courbure n'est pas définie. L'objectif consiste donc à approcher localement le maillage par une surface lisse pour calculer la courbure avec les formules précédentes. Parmi les nombreux travaux qui se sont penchés sur le problème, on distingue trois grands types de méthodes.

La première consiste à ajuster localement une surface analytique (généralement une quadrique) sur un petit voisinage des sommets du maillage original (on parle alors de *fitting*) [Ham93, CP03]. Cette méthode produit des résultats exacts lorsque

la forme des sommets correspond bien à la surface analytique utilisée. Par contre, elle provoque des erreurs lorsque la surface est dégénérée. [GI04] montre qu'il est possible d'éviter ces erreurs en utilisant des surfaces analytiques complexes d'ordre trois et en utilisant les normales en plus de la position des sommets.

Une autre manière de procéder consiste à calculer la courbure dans plusieurs directions à partir d'un sommet donné puis d'ajuster une quadrique sur les données résultantes (avec par exemple la méthode des moindres carrés) pour en déduire le tenseur [Tau95, PKS⁺01, HS02, MDSB02]. La précision et la robustesse des résultats obtenus dépendent ici aussi de la fonction adaptée sur les données extraites.

Enfin, il est possible d'obtenir le tenseur avec des méthodes qui effectuent une moyenne des tenseurs calculés sur chacune des faces pour parvenir à un résultat correct et lisse sur l'ensemble du maillage [CSM03, ACS⁺03]. Rusinkiewicz [Rus04] utilise une approximation par différences finies, en évitant que des erreurs se produisent sur les maillages irréguliers. Il existe évidemment de nombreuses autres méthodes qui permettent d'améliorer le calcul du tenseur. Par exemple, les angles des jonctions sur des objets ou des squelettes sont pris en compte dans [KNS⁺09] pour permettre de précalculer les courbures sur un petit intervalle d'animation. L'obtention du tenseur se fait même presque en temps-réel dans [BW07] avec l'utilisation du processeur graphique (GPU). Pour des informations plus précises sur la manière de calculer un tenseur à la surface d'un objet 3D, le lecteur peut se référer à [RDF05, BPK⁺07].

L'utilisation de toutes les informations contenues dans le tenseur n'est pas toujours nécessaire suivant l'application et le style dans lesquels on souhaite l'intégrer. D'autres informations sont souvent privilégiées pour leur plus grande intuitivité dans la description de la forme. La courbure gaussienne $K = \kappa_1 \kappa_2$, égale au déterminant de \mathbf{II} , permet par exemple de déterminer facilement si la surface est ombilique ou si elle a plutôt la forme d'une selle lorsqu'elle est respectivement positive ou négative. La courbure moyenne $H = (\kappa_1 + \kappa_2)/2$, qui correspond à la trace de la matrice, est un peu plus intuitive car elle permet de différencier les zones convexes ($H > 0$) ou concaves ($H < 0$) et les régions planes ($H = 0$).

4.2 Descripteurs hybrides

Ces informations calculées à la surface des objets ne sont pas toujours suffisantes car notre perception de la courbure peut se baser sur des données différentes [FTA04]. Dans de nombreuses applications, il est en effet important de prendre le point de vue en compte pour calculer ce que l'on peut *voir* depuis une caméra plutôt que ce que l'on *sait* de la géométrie de l'objet. La courbure centrée sur le point de vue diffère sur certaines régions de surface (notamment sur les zones proches des silhouettes) et devrait aussi être modifiée lorsque les objets sont proches ou éloignés de la caméra pour obtenir des niveaux de détail pertinents.

4.2.1 Courbure apparente [JDA07]

Judd *et al.* [JDA07] montrent par exemple que, pour calculer les crêtes apparentes sur des objets 3D, la courbure d'une sphère n'est pas la même partout lorsque la caméra est prise en compte (cf. Figure 4.2). En analysant la variation des normales dans l'espace écran, la courbure est plus grande lorsqu'on se rapproche des silhouettes qu'au centre de la sphère. Pour déterminer cette information dépendante du point de vue, Judd *et al.* projettent le tenseur de courbure (précalculé grâce à l'une des méthodes vues précédemment) sur l'écran. Néanmoins, cette technique souffre d'un certain nombre de limitations.

Premièrement, le tenseur résultant n'est plus symétrique et les vecteurs propres projetés ne sont plus orthogonaux ; seule la direction (et l'intensité) de courbure maximale est alors utilisable dans leur application. De plus, leur calcul souffre aussi du fait qu'il est effectué à l'aide d'un voisinage fixe sur le maillage des objets. Il n'est donc pas possible d'obtenir des niveaux de détail automatiques pour le descripteur lorsque les objets de la scène s'approchent ou s'éloignent de la caméra, à moins d'avoir recours à des techniques assez lourdes de simplification de maillages à plusieurs échelles [NJLM06, JNLM05].

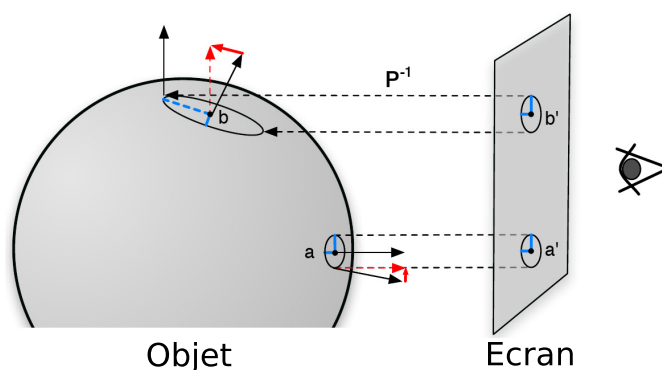


Fig. 4.2: La courbure calculée à la surface d'une sphère est uniforme. En la paramétrant dans l'espace écran, les domaines de calcul des dérivées sont uniformes sur le plan mais ne le sont plus après leurs projections sur l'objet ; cela a pour effet d'accroître la valeur de courbure le long des silhouettes [JDA07].

4.2.2 Descripteur de relief apparent

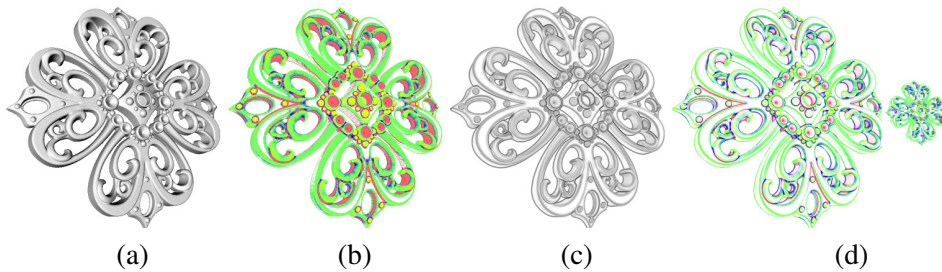
Pour éviter une telle surcharge d'informations dans le message, il est important d'intégrer directement le point de vue dans le calcul de la courbure. Dans cette section, nous introduisons une première approche pour prendre en compte le point de vue ainsi que les niveaux de détail lors du calcul d'un descripteur de relief apparent. Ce descripteur fait partie de l'article publié à NPAR [VBGS08a]. La méthode

consiste à modifier le tenseur original \mathbf{II} (que nous calculons avec la méthode de Rusinkiewicz [Rus04]) en y intégrant une valeur de courbure dépendante du point de vue, tout en conservant les directions de courbures principales originales. L'objectif est donc de modifier les valeurs de courbures qui seront issues du tenseur, sans altérer les directions.

Nous calculons cette valeur de courbure, que nous appelons L , en utilisant les normales des objets projetées en espace écran, de sorte à prendre le point de vue en compte et à fournir des niveaux de détail automatiques. Notre solution consiste à calculer la variation des normales suivant les deux axes \mathbf{x} et \mathbf{y} du plan de l'image, puis de les combiner pour obtenir L (cf. Figure 4.3). Pour cela, nous considérons un voisinage étendu et nous convoluons les deux premières coordonnées des normales n_x et n_y projetées en espace image avec des dérivées de Gaussiennes :

$$\begin{aligned}\nabla n_x(p) &= \begin{pmatrix} n_x(p) \otimes g_x(p, \sigma) \\ n_x(p) \otimes g_y(p, \sigma) \end{pmatrix} \\ \nabla n_y(p) &= \begin{pmatrix} n_y(p) \otimes g_x(p, \sigma) \\ n_y(p) \otimes g_y(p, \sigma) \end{pmatrix}\end{aligned}$$

où $g_x(p, \sigma)$ et $g_y(p, \sigma)$ correspondent respectivement aux noyaux de dérivées de Gaussiennes dans les directions \mathbf{x} et \mathbf{y} , et σ détermine l'échelle à laquelle la dérivée est calculée. Nous utilisons la séparabilité de l'opérateur de dérivée de Gaussi-



F . 4.3: Extraction du descripteur de relief apparent. Le rouge et le vert correspondent respectivement aux concavités et convexités. Le bleu représente les transitions entre ces deux types de surface (les T-jonctions). L'intensité des couleurs représente l'intensité de la courbure. Étant donné un objet 3D en entrée (a), nous commençons par analyser les attributs en espace objet pour récupérer l'information de convexité (b). Ensuite, nous regardons la variation des normales en espace image pour en déduire une intensité de courbure dépendante du point de vue et de la résolution (c). Ces deux caractéristiques sont combinées pour obtenir le descripteur de relief apparent (d). On peut noter les niveaux de détail automatiques lorsque l'objet s'éloigne du point de vue.

ennes pour calculer la variation des normales rapidement sur le GPU : $g_x(p, \sigma) = g'(p_x, \sigma)g(p_y, \sigma)$ et $g_y(p, \sigma) = g(p_x, \sigma)g'(p_y, \sigma)$ (voir [tHR03] pour plus de détails).

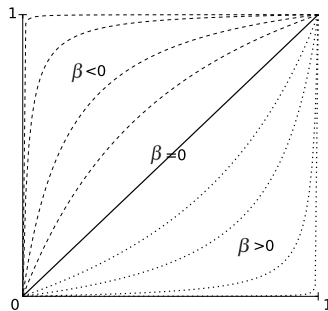
Nous pouvons maintenant extraire l'information de courbure à partir de ces deux gradients. Nous nous sommes inspirés pour cela des travaux qui tentent de trouver le gradient le plus représentatif sur des images composées de plusieurs canaux. Dans [Zen86], Zenzo présente une méthode générale pour calculer des gradients de couleurs à partir de triplets $(\nabla R, \nabla V, \nabla B)$. Nous utilisons cette technique sur nos deux gradients de normales $(\nabla n_x, \nabla n_y)$. Elle consiste à combiner les gradients directionnels pour les intégrer dans un tenseur symétrique :

$$T = \begin{pmatrix} \nabla_x n \cdot \nabla_x n & \nabla_x n \cdot \nabla_y n \\ \nabla_y n \cdot \nabla_x n & \nabla_y n \cdot \nabla_y n \end{pmatrix} \text{ avec } \nabla_x n = \begin{pmatrix} \nabla n_x \cdot \mathbf{x} \\ \nabla n_y \cdot \mathbf{x} \end{pmatrix} \text{ et } \nabla_y n = \begin{pmatrix} \nabla n_x \cdot \mathbf{y} \\ \nabla n_y \cdot \mathbf{y} \end{pmatrix}$$

Il suffit enfin d'extraire de la matrice T le vecteur propre dont la valeur absolue est maximum pour obtenir l'intensité de courbure L . Nous modifions aussi l'information de courbure avec une tangente hyperbolique de sorte que la valeur L soit comprise dans l'intervalle $[0, 1]$; elle est ainsi facilement exploitable. Toutes ces opérations sont facilement implémentables dans le *fragment shader* des cartes graphiques actuelles. Avec l'utilisation seule d'une carte de normales en entrée, L peut être facilement calculée en temps-réel sur des scènes dynamiques.

La dernière étape de cette technique consiste à intégrer cette valeur de courbure au tenseur original. Ceci se fait très simplement en multipliant \mathbf{II} de sorte que la valeur absolue de la courbure moyenne soit égale à L :

$$\mathbf{II}' = \frac{L}{H} \mathbf{II}$$



F . 4.4: En jouant sur le paramètre β , il est possible d'intensifier ou d'atténuer l'effet de la courbure dans le descripteur, tout en conservant un intervalle compris entre 0 et 1.

Contrairement à l'approche de Judd *et al.* [JDA07], le tenseur que nous obtenons reste symétrique et intègre une information dépendante du point de vue. Nous laissons aussi à l'utilisateur la possibilité de contrôler l'intensité de courbure. Nous avons pour cela mis en place une fonction $\mathcal{E}_\beta(x)$ qui permet d'accentuer ou d'atténuer les valeurs du tenseur. Elle est basée sur la fonction rationnelle de Schlick [Sch94b] : $\mathcal{E}_\beta(x) = x/(e^\beta(1-x) + x)$ (cf. Figure 4.4).



Figure 4.5: Notre information de dérivée permet d'extraire les détails caractéristiques d'une surface. Il est aussi possible de concentrer la vue sur certaines régions (comme ici la patte du dinosaure) en variant le niveau de détail par pixel.

Le paramètre de lissage σ peut aussi être utilisé pour contrôler la largeur des détails qui sont extraits. Deux approches différentes peuvent être considérées : l'approche globale dans laquelle σ est constant sur toute l'image, et une approche plus locale dans laquelle il est possible de faire varier σ en fonction de différents paramètres comme la profondeur ou la distance à la souris (cf. exemple de la Figure 4.5). Dans ce dernier cas, il suffit de calculer L à des échelles multiples, puis de les combiner linéairement.

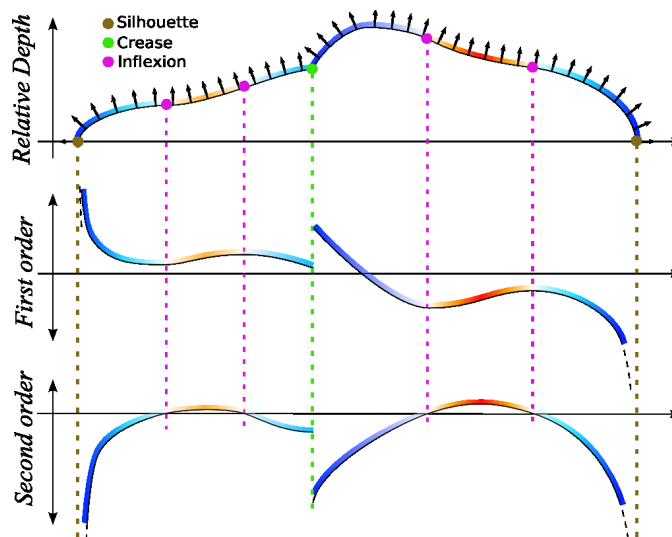
Si ce descripteur a l'avantage de prendre le point de vue en compte et de procurer des niveaux de détail automatiques, il souffre aussi de certaines limitations. Il n'est par exemple pas très bien adapté aux scènes dynamiques car il serait nécessaire de recalculer le tenseur en espace objet à chaque image. Les niveaux de détail ne sont pas non plus parfaits, surtout lorsque l'objet se trouve loin de la caméra ou que l'échelle est élevée. Malgré le lissage effectué, les courbures principales initialement calculées à la surface du maillage s'adaptent mal aux différentes échelles. En effet, les deux informations de direction et de courbure ne se combinent pas toujours très bien et peuvent provoquer des incohérences spatiales, notamment autour des silhouettes. L'intensité de courbure élevée sur ces zones peut déborder et mettre en avant deux types de surface différentes : celle de la partie occultée et celle de la partie occultante.

4.3 Descripteur centré sur le point de vue

Pour éviter ces problèmes, nous avons développé un descripteur local de forme, entièrement dépendant du point de vue et basé uniquement sur deux informations : les normales projetées en espace image et une carte de profondeurs des objets. Ainsi, nous pouvons extraire un tenseur de courbure de manière complètement dynamique et en temps-réel. Pour cela, nous prenons exemple sur Toler-Franklin *et al.* [TFFR07] qui montrent comment calculer un tenseur sur des images fixes composées des informations de couleur et de normale. Ce descripteur fait partie de l'article [VPB⁺09] publié à la conférence SIGGRAPH.

4.3.1 Cas 1D

Revenons sur le cas 1D d'un champ de normales pour comprendre le procédé. Comme on peut le voir sur la première ligne de la Figure 4.6, les normales définies depuis un point de vue donné définissent implicitement les variations de hauteur des points à la surface des objets. Les dérivées premières et secondes de ce champ de hauteurs sont tracées respectivement sur la ligne du milieu et celle du bas. On remarque notamment que les silhouettes sont des singularités sur lesquelles le champ de hauteurs n'est pas dérivable. En ces points, la pente de la courbe (définie par les normales) devient perpendiculaire au point de vue et la dérivée en espace image devient infinie. Les arêtes vives sont aussi des singularités du premier ordre ; dans ce cas, les normales varient brutalement. Ces singularités représentent les frontières



F . 4.6: Cas 1D : les normales définissent implicitement un champ de hauteurs avec une information de profondeur relative. Les dérivées successives de ce champ révèlent des singularités comme les silhouettes ou les arêtes vives et les informations de courbure souhaitées.

entre différentes régions à la surface de l'objet. Les points d'inflexion représentent aussi une caractéristique importante de l'analyse. Définis comme étant les maxima de la dérivée première et les changements de signe de la dérivée seconde, ils séparent les régions convexes des régions concaves. Entre les silhouettes, les arêtes vives et les points d'inflexions, la dérivée seconde donne des informations sur la courbure de la surface centrée sur le point de vue. Le signe de la courbure dépend de la manière dont les normales sont orientées sur la surface. Dans le cas d'un maillage, elles sont orientées vers la caméra ; il en résulte une courbure négative (en bleu) pour les convexités et une courbure positive (en rouge) pour les concavités.

4.3.2 Analyse de la variation des normales en 2D

La méthode de calcul du tenseur en espace image est similaire à celle utilisée pour le maillage. Il y a donc une relation directe entre la dérivée du champ de hauteurs et la normale projetée en espace écran. Notons \mathbf{x} , \mathbf{y} et \mathbf{z} les axes du plan image et $\mathbf{n}(p) = (n_x, n_y, n_z)$ les coordonnées de la normale située sur le point p de l'image. Le gradient de la profondeur relative $d(p)$ est alors donné par :

$$\mathbf{g}(p) = \nabla d(p) = \begin{pmatrix} \frac{\delta d(p)}{\delta x} \\ \frac{\delta d(p)}{\delta y} \end{pmatrix} = \begin{pmatrix} -n_x/n_z^f \\ -n_y/n_z^f \end{pmatrix}$$

où f , qui en théorie est égal à 1, est un paramètre utilisateur $f \in [0, \infty[$ qui permet de contrôler l'intensité de la courbure lorsqu'on se rapproche des silhouettes. Sur ces régions, n_z tend vers 0 et le gradient de la profondeur est indéfini. Comme dans les travaux de Judd *et al.* [JDA07], nous considérons que la projection est localement orthogonale et que la paramétrisation en espace image est similaire à celle que l'on pourrait avoir en espace objet. Dans ce cas, la matrice hessienne correspond au tenseur de courbure ($\mathbf{H} = \mathbf{II}$) et est définie via la dérivée de ce gradient :

$$\mathbf{II} = \nabla^T \nabla d(p) = \nabla^T \mathbf{g}(p) = \begin{pmatrix} \frac{\delta \mathbf{g}}{\delta x} & \frac{\delta \mathbf{g}}{\delta y} \end{pmatrix} = \begin{pmatrix} e & f \\ f & g \end{pmatrix}$$

Il ne reste plus qu'à extraire les vecteurs et valeurs propres de cette matrice 2×2 symétrique pour obtenir les directions et courbures principales, comme on l'a vu dans la Section 4.1. \mathbf{II} n'étant pas défini sur les arêtes vives et les silhouettes, il nous faut aussi prendre en compte ces éléments dans l'implémentation. Néanmoins, nous considérons la plupart du temps que la surface des objets est lisse (et donc dérivable). Notre descripteur local de forme consiste donc en l'union des silhouettes, des arêtes vives et des autres points de la surface sur lesquels nous calculons \mathbf{II} . Dans l'exemple de la Figure 4.7, les silhouettes et les arêtes vives sont affichées en noir. Les régions concaves et convexes sont affichées à l'aide de la courbure moyenne $H = (\kappa_1 + \kappa_2)/2$ avec des couleurs respectivement chaudes et froides comme le montre l'échelle des niveaux de couleur sur la droite.

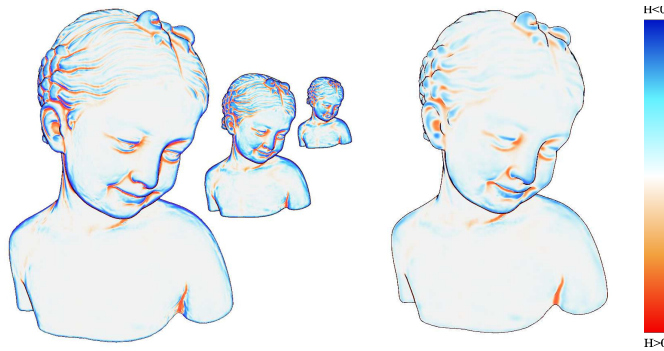


Fig. 4.7: À gauche : calculé en espace image, le descripteur se simplifie automatiquement lorsque l'objet s'éloigne de la caméra et ne sélectionne que des détails assez larges. À droite : nous pouvons aussi varier l'échelle du descripteur sur chacun des pixels pour focaliser l'observation sur une zone (ici autour de la bouche de l'enfant) et obtenir des niveaux de détail intéressants. Le modèle provient de ©AIM@SHAPE.

4.3.3 Sélection du niveau de détail

L'un des avantages majeur de notre descripteur est qu'il permet d'obtenir des niveaux de détail automatiques sur les objets lorsqu'ils s'approchent ou s'éloignent de la caméra (cf. Figure 4.7). De plus, la méthode fonctionne en temps-réel avec seulement les cartes de normales et de profondeurs en entrée ; elle permet donc d'être facilement appliquée sur des objets déformables, avec les détails de surface pouvant être donnés par une carte de normales, et tout ceci, sans précalcul. Comme dans le cas du descripteur de relief apparent, nous souhaitons que l'utilisateur puisse sélectionner le niveau de détail qui lui convient ou même de faire varier l'échelle sur l'image dans le but d'augmenter l'attention sur certaines parties de la scène. Pour éviter le problème de débordement sur les singularités (silhouettes ou arêtes vives) obtenu avec le vecteur de forme dans la section précédente, nous intégrons les normales projetées avec une diffusion anisotrope [PM90] :

$$\frac{\delta \mathbf{g}_s(p)}{\delta s} = \nabla \cdot (c(p) \nabla \mathbf{g}_s(p))$$

où s représente l'échelle à laquelle nous souhaitons diffuser le gradient et $c(p)$ est la fonction de conductance, qui dans notre cas est égale à 0 sur les silhouettes et les arêtes vives et à 1 partout ailleurs. Définie ainsi, la conductance bloque la diffusion à travers ces discontinuités. Nous appliquons la diffusion un certain nombre de fois dans le but de produire un gradient lissé \mathbf{g}_s qui préserve les silhouettes et les arêtes vives. Nous obtenons ainsi une approximation d'un tenseur de courbure dont les valeurs correspondent à des détails plus ou moins larges : $\mathbf{\Pi}_s(p) = \nabla^T \mathbf{g}_s$. Nous laissons aussi à l'utilisateur la possibilité de paramétrer spatialement le processus

de diffusion à l'aide d'une fonction d'importance $I(p)$ qui contrôle localement le nombre d'itérations. Un nombre d'itérations faible ou élevé permet respectivement d'extraire des détails fins ou larges. Comme pour le descripteur hybride, toute fonction d'importance peut être appliquée. Un exemple est donné dans la Figure 4.7 à droite, où des détails plus fins sont donnés autour de la bouche de l'enfant.

4.3.4 Implémentation

Notre descripteur local de forme est calculé par pixel, entièrement sur le GPU en quelques passes. Les seules données nécessaires en entrée sont les cartes de normales et de profondeurs. Elles sont obtenues très simplement en projetant et en rastérisant ces données lors de la première passe du *vertex au fragment shader*. Dans l'algorithme ci-dessous, p dénote le pixel courant et p_i son voisinage 3×3 . Le tenseur de second ordre $\mathbf{\Pi}_s$ est alors calculé par pixel comme décrit par l'algorithme suivant :

- 1: $w_s(p) \leftarrow$ Filtre de Sobel (Profondeur(p_i))
- 2: $w_c(p) \leftarrow$ Angle Dihédral ($\mathbf{n}(p_i)$)
- 3: $\mathbf{g}_0(p) \leftarrow$ Gradient de Profondeur ($\mathbf{n}(p)$)
- 4: **for** $s \in [1..I(p)]$ **do**
- 5: $\mathbf{g}_s(p) \leftarrow$ Diffusion Anisotrope ($\mathbf{g}_{s-1}(p_i), w_s(p), w_c(p)$)
- 6: **end for**
- 7: $\mathbf{\Pi}_s(p) \leftarrow$ Filtre de Sobel ($\mathbf{g}_s(p_i)$)

Cinq étapes principales sont décrites dans cet algorithme. Les silhouettes (1) sont tout d'abord calculées avec un filtre de Sobel appliqué sur les profondeurs pour obtenir un poids $w_s(p)$ par pixel. Empiriquement, cette approche se révèle être plus cohérente que de détecter les points exacts pour lesquels $n_z = 0$. Les arêtes vives (2), dont les poids sont égaux à $w_c(p)$, sont obtenues en analysant l'angle dihédral entre la normale au point p et les normales voisines. Le gradient multi-échelles \mathbf{g}_s est extrait en calculant \mathbf{g}_0 (3) et en appliquant l'équation de diffusion anisotrope (4-6) de manière itérative comme expliqué dans [PM90]. Nous utilisons $c(p) = 1 - \max(w_s, w_c)$ pour la fonction de conductance. Le tenseur est enfin obtenu lors d'une dernière étape (7) en calculant la dérivée de ce gradient avec un filtre de Sobel. La courbure n'étant pas définie sur les silhouettes et les arêtes vives, nous utilisons $c(p)$ pour interpoler linéairement entre $\mathbf{\Pi}_s(p)$ et une matrice nulle (qui correspond à une région plane) sur les singularités.

4.4 Comparaison des descripteurs

D'un point de vue pratique, le descripteur local de forme que nous venons de voir offre de nombreux avantages. Contrairement au tenseur calculé en espace objet ou au descripteur hybride, il n'est pas dépendant de la complexité de la scène. Il est extrait en parallèle sur chaque pixel grâce aux processeurs graphiques actuels.

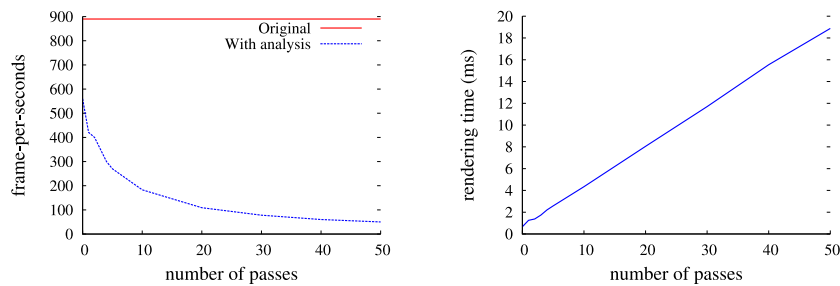


FIG. 4.8: Impact du descripteur local sur le temps de rendu. A gauche : le taux de rafraîchissement décroît avec l'augmentation du nombre de passes de diffusion. A droite : coût du processus lors de l'augmentation du nombre de passes. Chaque étape de diffusion ajoute un surplus de 0.4 ms. Ces tests ont été faits avec un processeur Intel(R) Core(TM)2 Quad à 2.4 GHz, une carte graphique NVIDIA GeForce 8800 GTX et une image de résolution 800×600 .

Il se calcule donc en temps-réel et sa complexité augmente linéairement avec le nombre d'itérations utilisées pour la diffusion (cf. Figure 4.8) et avec le nombre de pixels dans l'image. Le descripteur objet peut se révéler pratique pour analyser la surface sur des scènes fixes car il suffit de précalculer le tenseur en chacun des sommets pour l'intégrer dans une application ou pour y ajouter des informations dépendantes du point de vue comme dans notre descripteur hybride. Le descripteur local se révèle très utile pour les objets dynamiques et déformables. Il peut être appliqué sur n'importe quel type de données car il ne prend en entrée que les textures de normales et de profondeurs. Il peut donc être calculé facilement sur des objets modifiés par des cartes de normales, des surfaces composées de points ou même des images RGBN [TFFR07] (composées d'une image et d'une carte de normales). Enfin, ce dernier descripteur garde l'avantage de prendre en compte le point de vue lors de l'analyse. Il permet aussi de créer des niveaux de détail automatiques et contrôlables de manière cohérente. La Figure 4.9 montre les différences obtenues avec le descripteur centré sur l'objet (a) et le descripteur centré sur le point de vue (b-d). Le paramètre f ajouté lors du calcul du gradient de la profondeur permet de contrôler l'intensité de la courbure sur les régions proches des silhouettes.

Le principal problème engendré par le descripteur local dépendant du point de vue est qu'il nécessite des surfaces lisses pour fournir un résultat correct. Comme on peut le voir sur la Figure 4.10 qui a été faite après un zoom sur l'œil de l'enfant de la Figure 4.7, notre descripteur commence à mettre en valeur la tessellation de la géométrie elle-même. Cela peut être vu comme une limitation car des détails non désirables apparaissent, avec notamment les transitions d'intensité de courbures abruptes entre les triangles. Néanmoins, cet effet est logique car la continuité des normales à la surface de l'objet est seulement C^0 le long des arêtes des triangles. Cela est dû à l'interpolation de Phong implémentée sur la surface lors de

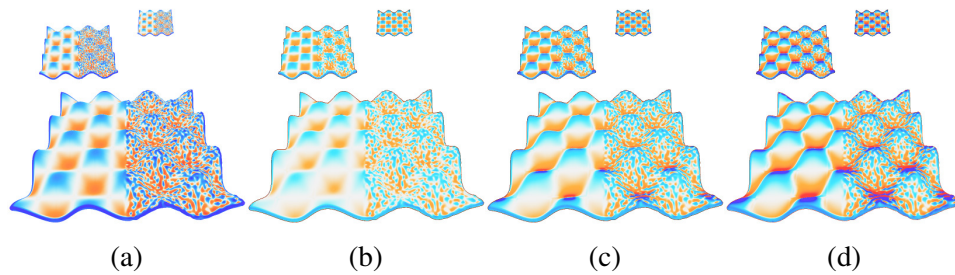


Fig. 4.9: (a) Mesure de la courbure moyenne avec une méthode complète en espace objet. Les détails de surface, aussi petits soient-ils, sont conservés, même lorsque l'objet est situé loin de la caméra. (b-d) Avec la courbure moyenne extraite du descripteur centré sur le point de vue, les détails de surface apparaissent et disparaissent progressivement suivant la taille de leur projection dans l'image. Le paramètre f permet aussi de contrôler l'intensité de courbure sur les régions proches des silhouettes : (b) $f = 0$; (c) $f = 0.5$; (d) $f = 1.0$.

la rasterisation sur la carte graphique. Une manière simple de résoudre ce problème est d'appliquer une subdivision dynamique sur le maillage [BS05, BRS05] de sorte que chacun des triangles se retrouve à l'intérieur d'un pixel dans l'image finale. Une autre manière de procéder est d'adapter l'échelle de notre descripteur en prenant en compte la profondeur de la surface et ainsi lisser les surfaces qui ne sont pas assez tessellées. Enfin d'autres types de surfaces peuvent être utilisés sans problèmes, comme les surfaces implicites ou les surfaces à base de points [BSK04, GBP06, WHA⁺07].

Une autre limitation est l'aliasing des normales qui peut engendrer des incohérences dans le calcul de l'information de courbure. Les normales ne sont pas forcément bien interpolées lorsque les objets s'éloignent de la caméra et des scintillements peuvent apparaître. Une solution possible consiste à utiliser des techniques de lissage adaptées [OB10]. De plus, il est raisonnable de penser que les cartes graphiques futures intégreront directement des méthodes d'anti-aliasing de qualité.

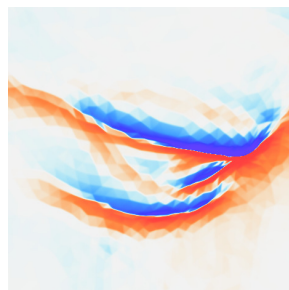


Fig. 4.10: Effet du descripteur sur des surfaces très peu tessellées.

Modifications de l'équation de réflexion

Nous disposons désormais d'une description de la surface en chacun des points de la scène. L'objectif dans ce chapitre est de montrer différentes manières d'intégrer cette information dans le rendu, et notamment grâce à l'équation de réflexion, pour créer des styles variés et contrôlables. Chacune des techniques que nous

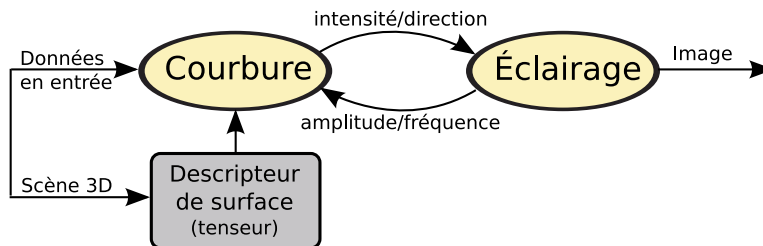


Fig. 5.1: Le message extrait les paramètres nécessaires depuis une description de la surface (dans notre cas un tenseur de courbure). Ces informations sont utilisées pour exagérer les détails dans une grande variété de styles.

allons introduire entretient des liens différents entre le message et le style. Néanmoins, elles ont toutes l'utilisation du tenseur de courbure comme point de départ. Le message que nous utilisons ici est donc une donnée issue du tenseur de courbure (cf. Figure 5.1). Le style fait, quant à lui, partie intégrante de l'équation de réflexion. Nous proposons trois manières différentes pour modifier l'équation (cf. Figure 5.2), chacune ayant pour objectif de mettre en valeur les détails de surface des objets 3D avec une grande variété de styles. La plus simple consiste à utiliser notre information de courbure pour modifier directement la radiance réfléchie, obtenue lors du calcul de l'image finale, dans le but de faire ressortir les détails de surface (a). La seconde est basée sur une technique qui permet d'améliorer notre perception de la forme pour produire des effets plus subtils, en jouant sur la fréquence de

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_{\mathbf{n}}} \rho(\mathbf{e}, \boldsymbol{\ell}) (\mathbf{n} \cdot \boldsymbol{\ell}) L(\mathbf{p} \leftarrow \boldsymbol{\ell}) d\boldsymbol{\ell}$$

Fig. 5.2: Nous modifions le calcul de l'énergie réfléchi à plusieurs endroits dans le but de mettre les détails de la scène en valeur. (a) Modification globale de la radiance réfléchi. (b) Modification de la radiance incidente. (c) Modification de la radiance réfléchi pour chaque direction des sources lumineuses.

l'éclairage incident (b). Nous verrons enfin comment modifier efficacement l'intensité lumineuse réfléchi de chacune des directions incidentes en fonction des *BRDFs* (c) afin d'obtenir plus de flexibilité dans les styles.

5.1 Modification de l'intensité globale

La première méthode que nous présentons consiste à modifier l'intensité globale pour chacun des pixels d'une image, en fonction de notre descripteur de forme dépendant du point de vue. Comme on peut le voir dans la Figure 5.2 (a), cela revient à ajouter un terme multiplicateur en dehors de l'intégrale dans l'équation de réflexion. Modifier ainsi l'intensité de manière globale, indépendamment de la radiance incidente et de la réponse du matériau, apporte des contraintes dans la manière de mettre les détails en valeur (on ne connaît ni les matériaux, ni les sources lumineuses), mais aussi une grande flexibilité dans le style utilisé. N'importe quelle fonction de rendu peut en effet être utilisée, qu'elle soit réaliste ou non. Le message qu'il est possible de transmettre est donc limité à l'utilisation directe de l'information de courbure pour modifier la couleur du rendu. Les directions et intensités de courbure principales n'étant pas directement exploitables dans ce cas, il nous faut résumer le tenseur à un scalaire.

5.1.1 Contrôle des formes avec une carte de relief apparent

Le vecteur de forme de Koenderink et van Doorn [KvK92] est parfaitement adapté dans ce cas. Il s'agit d'une représentation plus intuitive contenant une information complète sur la forme de la surface. Ce vecteur, noté \mathbf{S} , est créé à partir des seules courbures principales κ_1 et κ_2 normalisées et permet de représenter facilement tout type de surface dans un domaine 2D (cf. Figure 5.3 (a,b)) : la direction du vecteur donne des informations sur la convexité de la surface tandis que sa longueur renseigne sur l'intensité de courbure. On peut remarquer la symétrie qui apparaît autour de la diagonale dans la Figure 5.3 ; elle est due aux valeurs de κ_1 et κ_2 qui correspondent arbitrairement aux courbures minimales ou maximales.

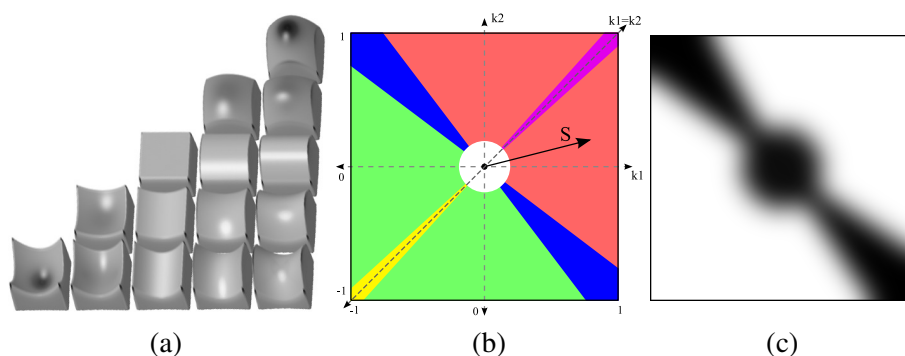


Fig. 5.3: (a) Les différentes formes de surface qu'il est possible de représenter dans le domaine du vecteur de forme de Koenderink et van Doorn [KvK92] (b). Les axes correspondent aux courbures principales et permettent de différencier tout type de surface : les régions planaires (en blanc), et les formes localement sphériques concaves (en jaune) ou convexes (en violet) sont situées autour de l'axe de symétrie. La forme en selle (en bleu) sépare les régions concaves (en vert) des régions convexes (en rouge). (c) une carte de relief apparent est une texture en niveau de gris permettant de sélectionner le type de forme que l'on souhaite exagérer. Dans cet exemple, les surfaces planaires (au centre) et les selles (sur la diagonale) sont représentées en noir et filtrées lors du rendu. Les autres types de surface (convexités, concavités, etc.) seront mis en valeur.

Lorsqu'il est extrait depuis l'un des descripteurs que nous avons introduit dans le Chapitre 4 (descripteur hybride ou descripteur local de forme), ce vecteur conserve les propriétés qui peuvent prendre le point de vue en compte. Cette méthode a notamment été publiée à la conférence NPAR [VBGS08a, VBGS08b].

L'avantage principal que procure ce vecteur est que, grâce à sa paramétrisation 2D, il est possible d'utiliser une simple texture pour contrôler les différentes caractéristiques de la surface que l'on souhaite mettre en avant (cf. Figure 5.3 (c)). Par ce moyen, nous assignons une valeur de *relief* $r \in [0, 1]$ pour chacun des vecteurs de forme possible et ainsi nous contrôlons les formes mises en valeur. En disposant d'un vecteur de forme sur chacun des pixels de l'image, il suffit de chercher la valeur r correspondante avec un simple accès dans la texture de relief apparent. Dans l'exemple de la Figure 5.3 (c), tous les types de surface ont été sélectionnés (lorsque $r = 1$), excepté les régions planaires et les formes de selles (avec $r = 0$). Nous proposons deux méthodes pour éditer cette texture dans notre prototype. En dessinant directement dans la texture, l'utilisateur peut voir les détails révélés sur l'objet 3D en temps-réel. Il est aussi possible d'utiliser le procédé inverse, en sélectionnant directement le type de forme à exagérer à la surface de l'objet, puis en mettant la texture à jour.

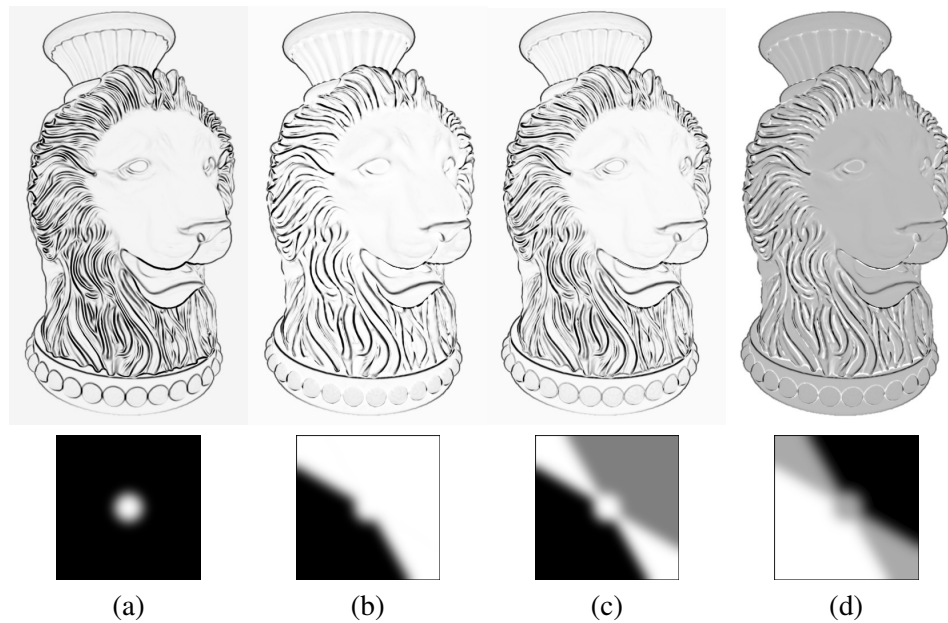


Fig. 5.4: (a) Toute la surface est affichée en noir, excepté les régions planes. (b) Seules les convexités sont affichées en noir. (c) Résultat intermédiaire entre (a) et (b) dans lequel les concavités sont affichées en gris. (d) Les concavités sont affichées en blanc, les convexités en noir et les zones planes ainsi que les selles sont représentées en gris (0.4 M triangles / 244 fps).

5.1.2 Styles

La valeur de relief extraite de la texture pour chacun des pixels dans l'image peut être combinée avec le rendu de différentes manières, suivant le style utilisé. Nous montrons ici quelques styles se rapprochant de ceux utilisés dans les bandes dessinées pour illustrer l'intégration de la forme dans le rendu. Les données issues de l'analyse de la surface que nous utilisons dans cette section proviennent du descripteur hybride présenté dans la Section 4.2. Néanmoins, il serait possible d'utiliser de la même manière la courbure centrée sur l'objet ou sur le point de vue (cf. Chapitre 4).

Style crayonné

La manière la plus simple de faire apparaître les détails sur les objets 3D est d'utiliser directement la valeur de relief r comme couleur dans le rendu. La Figure 5.4 montre différents rendus dans lesquels les couleurs varient en fonction du type des surfaces. La ligne du haut montre le résultat obtenu avec l'utilisation de la texture de relief apparent correspondante sur la ligne du bas. Par exemple, tous les détails sont dessinés en noir, excepté les surfaces planaires dans l'exemple de

gauche. Les trois autres images montrent des effets plus subtils pour différencier certains types de surface (les convexités, concavités, selles, etc.).

Style dessin animé

Les styles cartoon ont l'avantage de simplifier les rendus avec seulement quelques couleurs. Néanmoins, les aplats générés peuvent masquer de nombreux détails importants dans les images finales. Pour améliorer cela, nous utilisons *X-toon* [BTM06], introduit dans le Chapitre 3. Pour rappel, cette technique utilise une texture 2D pour représenter toutes les couleurs dans des rendus stylisés. L'axe des abscisses représente l'intensité Lambertienne. La seconde dimension permet de modifier chacune des couleurs en fonction d'une valeur de détail, qui dans notre cas, sera la valeur de relief. Sur la Figure 5.5, on peut voir différents styles obtenus avec leurs textures *X-Toon* correspondantes. Les effets peuvent être multiples, suivant que l'on utilise des variations douces ou discontinues dans les couleurs, ou que l'on joue avec la transparence (avec le canal alpha). Nous montrons aussi qu'il est possible d'appliquer cette technique sur des objets animés sur la ligne du bas.

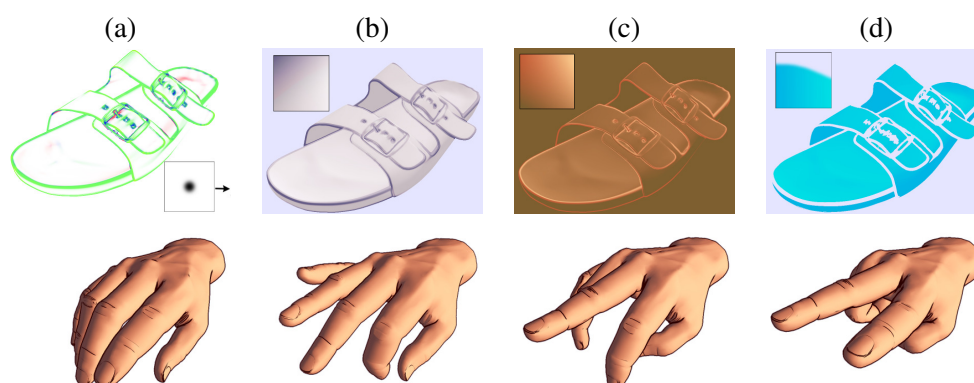


FIG. 5.5: (a) Le vecteur de forme obtenu sur chacun des pixels de la sandale permet de récupérer une valeur de relief (identifiant les convexités et concavités dans ce cas), elle-même utilisée pour sélectionner une couleur dans des textures *X-Toon*. (b) Des variations de couleurs douces sont utilisées pour faire apparaître les détails de manière progressive. (c) Les régions claires et sombres ont été inversées dans le but de montrer les variations à l'intérieur même des régions importantes (i.e. sélectionnées par la valeur de relief). (d) En choisissant des variations de couleurs assez faibles, et en faisant disparaître les détails grâce à la valeur alpha de la texture lorsque la courbure est élevée, le style semble beaucoup plus “aplati”. La sandale est rendue à 250 *fps*, avec 0.3 M triangles. La ligne du bas montre un exemple similaire sur une main animée.

Style minimal

Comme nous avons pu le voir dans le Chapitre 3, certains styles peuvent être encore plus drastiques. C'est le cas du noir et blanc utilisé par exemple dans le film d'animation "Renaissance" [Vol06] pour créer une ambiance sombre. Pour éviter que les artistes aient à manipuler les lumières pour faire apparaître les détails dans la scène, nous montrons comment réintroduire les caractéristiques de forme de manière continue avec la valeur de relief (Figure 5.6). L'idée est de considérer la valeur de relief comme une valeur de contraste pour inverser les couleurs sur les éléments importants. Dans les régions dessinées en noir, les détails seront rendus en blanc, et inversement dans les régions dont la couleur est blanche. Ce rendu minimal est donc défini comme étant une interpolation linéaire entre l'intensité diffuse et l'intensité diffuse inversée : $I = (1 - r)\mathbf{n} \cdot \mathbf{l} + r(1 - \mathbf{n} \cdot \mathbf{l})$. L'intensité I est seuillée à 0.5 pour obtenir l'effet noir et blanc. L'exemple de la Figure 5.6 montre la réintroduction des détails sur les parties concaves d'un objet.



FIG. 5.6: Rendu minimal (0.15 M triangles / 250 fps). Le descripteur de forme (a) permet de réintroduire le contraste perdu lors de l'utilisation d'un style aussi simple. Les régions convexes apparaissent ici comme blanc sur noir ou noir sur blanc suivant la couleur de rendu d'origine. Pour la portion de gauche, l'objet est éclairé par la droite. Au centre, nous tournons l'objet pour montrer les zones dans l'ombre. La source de lumière est ensuite orientée vers la gauche pour obtenir le résultat de droite.

Style cartoon avec cartes d'environnement

Nous combinons enfin l'information de courbure avec des rendus cartoon plus complexes. Dans l'exemple de la Figure 5.7, nous utilisons la courbure moyenne pour accroître le contraste entre convexités et concavités dans un rendu cartoon calculé à partir de nombreux échantillons dans une carte d'environnement :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \tanh(\alpha H)^\lambda L(\mathbf{p} \rightarrow \mathbf{e}) \quad (5.1)$$

où l'intensité de la courbure moyenne H est contrôlée avec le paramètre α . Le paramètre de contraste $\lambda \in [-1, 1]$ permet de laisser à l'utilisateur le choix d'as-

sombrir ou d'éclaircir les convexités et les concavités. La couleur finale est quantifiée avec la technique de Winnemöeller *et al.* [WOG06] pour créer des discontinuités entre les couleurs et donner l'effet cartoon.



FIG. 5.7: À gauche : un rendu cartoon original. À droite : les couleurs ont été éclaircies ou assombries avant la quantification pour mettre en valeur de nombreux détails avec le même nombre de couleurs.

5.2 Modification des rayons lumineux

L'un des principaux inconvénients de la modification globale de l'intensité est qu'il n'est pas possible de prendre en compte les informations concernant les matériaux, ou celles des radiances incidentes. La conséquence directe est que la mise en valeur des détails se fait nécessairement de la même manière sur tous les détails de forme sélectionnés. Pour mettre les détails en valeur dans des styles plus réalistes, sans perturber les propriétés des matériaux, nous proposons une autre solution, appelée *Light Warping* (LW), qui consiste à modifier la radiance incidente de chacune des sources lumineuses, localement et en tout point de la surface (cf. Figure 5.2 (b)). Cette méthode fait partie de l'article SIGGRAPH [VPB⁺09].

5.2.1 Motivations

Nous nous sommes directement inspirés des travaux de Fleming *et al.* [FTA04, FTA09] pour modifier la radiance entrante et mettre en valeur les détails quel que soit l'algorithme de rendu utilisé. Leurs travaux montrent que, si l'on ne prend pas les phénomènes de visibilité en compte, notre perception de la courbure est principalement due aux compressions et aux dilatations des motifs lumineux réfléchis à la surface des objets. Cet effet est illustré sur la Figure 5.8 (a), où l'on peut voir que les zones courbes contractent une plus grande partie de l'environnement lumineux que les zones planaires. Le motif produit dans l'image, depuis un point de vue donné, est donc plus compressé lorsqu'il est reflété par des surfaces courbes. Notre idée consiste donc à modifier les directions incidentes sur chacun des points de l'objet, de sorte à exagérer les effets de compressions et de dilatations produits, et donc à améliorer la perception de la courbure depuis un point de vue donné.

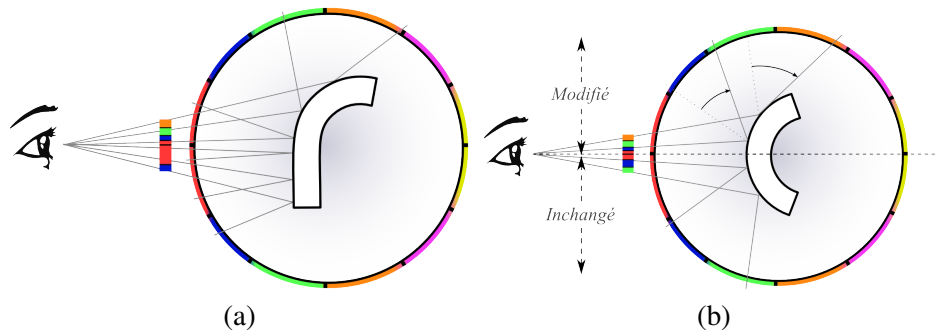


Figure 5.8: (a) Les compressions des motifs lumineux sur une surface parfaitement réfléchissante révèlent des informations sur la courbure depuis un point de vue donné. Les motifs réfléchis par les surfaces planaires représentent une petite partie de l'environnement. Les régions courbées réfléchissent quant à elles une plus grande partie de l'environnement, pour une même aire dans l'image observée. L'environnement reflété sur l'objet apparaît donc plus compressé sur les régions courbes ; cette propriété est utilisée par le Système Visuel Humain pour estimer la courbure. (b) En modifiant les directions des rayons lumineux, nous augmentons la région de l'environnement réfléchi sur l'objet afin d'accroître la perception de la forme.

Comme illustré sur la Figure 5.8 (b), cette modification se fait autour de la direction de vue, sur des détails de surface eux-mêmes dépendants du point de vue. L'utilisation de notre descripteur local de forme introduit dans la Section 4.3 du chapitre précédent s'avère ici nécessaire car il décrit les détails en se rapprochant des éléments pertinents pris en compte par le SVH [FTA04]. De plus, Fleming *et al.* [FTA09] montrent que ces effets de compression apparaissent le long des courbures principales ; nous verrons par la suite que notre tenseur nous permet de gérer cette propriété.

5.2.2 Modification de l'équation de réflexion

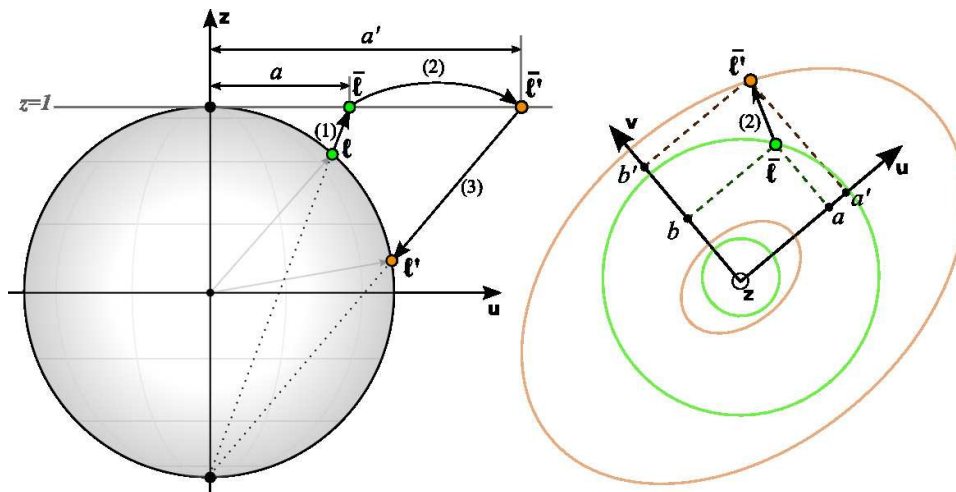
Le message que nous souhaitons communiquer consiste donc à corrélérer les convexités et concavités aux compressions et dilations des motifs lumineux réfléchis. En notant $W_{\mathbf{p}}(\boldsymbol{\ell})$ la fonction de modification de la direction incidente du rayon lumineux $\boldsymbol{\ell}$ au point \mathbf{p} , nous réécrivons l'équation de réflexion de la manière suivante :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_n} \rho(\mathbf{e}, \boldsymbol{\ell})(\mathbf{n} \cdot \boldsymbol{\ell}) L(\mathbf{p} \leftarrow W_{\mathbf{p}}(\boldsymbol{\ell})) d\boldsymbol{\ell} \quad (5.2)$$

La fonction W est une transformation bijective sur la sphère des directions, de sorte que la fonction inverse soit analytiquement définie. De plus, comme noté dans [FTA09], la compression des motifs lumineux est anisotrope et se fait le long

des directions de courbure principales. Nous prenons donc cette caractéristique en compte pour modifier la direction des lumières différemment le long des axes de courbures principales $\{\mathbf{t}_1, \mathbf{t}_2\}$. Notons $\{\mathbf{u}, \mathbf{v}, \mathbf{z}\}$ le repère local du plan image défini par les directions de courbures principales $\mathbf{u} = \mathbf{t}_1$ et $\mathbf{v} = \mathbf{t}_2$, \mathbf{z} étant la direction de vue, égale à $(0, 0, 1)$. Comme notre tenseur fournit des informations symétriques en espace image, la fonction W doit aussi être symétrique par rapport à \mathbf{u} et \mathbf{v} et invariante selon \mathbf{z} . Les directions des lumières n'étant pas définies dans ce repère, il est nécessaire de les projeter dans cette base avant de les modifier. Notre fonction de transformation est basée sur une projection stéréographique, de sorte à établir la correspondance entre l'espace cartésien du descripteur et celui angulaire de la direction lumineuse. Ce procédé, illustré par la Figure 5.9, est le suivant :

1. La direction ℓ de la lumière est tout d'abord projetée stéréographiquement sur le plan $z = 1$ pour donner $\bar{\ell} = S(\ell)$.
2. La direction de lumière $\bar{\ell}$ correspond à un point dans l'espace stéréographique, modifié par rapport aux courbures principales, donnant $\bar{\ell}' = W_S(\bar{\ell})$.
3. $\bar{\ell}'$ est re-projetée vers la sphère des directions et donne ainsi la direction de lumière modifiée $\ell' = S^{-1}(\bar{\ell}')$.



F . 5.9: À gauche : illustration 1D du procédé utilisé pour modifier les directions. La direction de lumière est (1) projetée stéréographiquement, (2) transformée dans une nouvelle position dans le plan stéréographique et (3), ré-exprimée dans la sphère des directions. A droite : illustration de l'étape (2) en 2D. On peut noter la symétrie autour des axes \mathbf{u} et \mathbf{v} .

Étant donné une direction de lumière $\boldsymbol{\ell} = (\ell_u, \ell_v, \ell_z)$, la projection stéréographique S est définie par :

$$S(\boldsymbol{\ell}) = (a, b, c) = \left(\frac{2\ell_u}{\ell_z + 1}, \frac{2\ell_v}{\ell_z + 1}, 1 \right)$$

La direction $(0, 0, 1)$ est projetée sur l'origine du plan stéréographique. Les autres projection de directions s'éloignent de l'origine jusqu'à la direction $(0, 0, -1)$, projetée à l'infini.

La fonction de modification des rayons est simplement définie comme une mise à l'échelle des positions lumineuses sur le plan stéréographique en fonction des courbures principales :

$$W_S(\bar{\boldsymbol{\ell}}) = (a', b', c') = (\lambda_u a, \lambda_v b, 1)$$

où les facteurs de mise à l'échelle $\lambda_{u|v}$ sont calculés en modifiant les courbures principales $\kappa_{u|v}$ par rapport à une déviation angulaire sur la sphère :

$\lambda_{u|v} = \tan(\arctan(\alpha\kappa_{u|v})/6 + \pi/4)$. Cela garantit qu'au maximum la moitié de l'énergie lumineuse se trouvant dans l'hémisphère est transmise à l'hémisphère opposé. À noter aussi que cette formulation est sensiblement similaire à celle utilisée pour la modification de la radiance globale présentée dans la section précédente. De la même manière, le paramètre α permet de laisser à l'utilisateur la possibilité de manipuler l'intensité de courbure et donc l'intensité de la modification effectuée sur les directions de lumière. Comme on peut le voir sur la Figure 5.9 (droite), l'anisotropie de courbure est naturellement prise en compte avec cette méthode.

La projection stéréographique inverse est enfin donnée par :

$$S^{-1}(\bar{\boldsymbol{\ell}}) = (\ell'_u, \ell'_v, \ell'_z) = (a't, b't, 2t - 1) \text{ avec } t = 4/(4 + a'^2 + b'^2)$$

où t localise l'intersection entre la sphère et la direction de projection. Ces trois opérations peuvent naturellement être formulées par une seule fonction de modification des rayons $W = S^{-1} \circ W_S \circ S$, donnée par :

$$W(\boldsymbol{\ell}) = \left(\frac{2t\lambda_u\ell_u}{1 + \ell_z}, \frac{2t\lambda_v\ell_v}{1 + \ell_z}, 2t - 1 \right), \quad t = \frac{(1 + \ell_z)^2}{(1 + \ell_z)^2 + \lambda_u^2\ell_u^2 + \lambda_v^2\ell_v^2}$$

La fonction inverse $W^{-1} = S^{-1} \circ W_S^{-1} \circ S$ est simplement obtenue en inversant λ_u et λ_v (i.e. en remplaçant α par $-\alpha$ dans le calcul de λ_u et λ_v).

5.2.3 Intégration dans différents styles

La fonction de modification W peut être utilisée avec n'importe quel style, du moment que l'on dispose d'un environnement suffisamment riche et échantillonné. Nous illustrons son utilisation dans cette section, avec des styles réalistes et des rendus cartoon, mais aussi avec des matériaux variés, allant de diffus à brillant.

Intégration dans un moteur d'éclairage global

Nous avons implémenté cette fonction de modification de rayons lumineux dans deux moteurs de rendus. Dans les deux cas, la *BRDF* d'Ashikmin [APS00] introduite dans le Chapitre 3, est utilisée.

Le premier d'entre eux est un système d'éclairage global utilisant la méthode de lancer de rayons [DBB06], dans lequel nous appliquons la méthode sur le premier rebond des rayons considérés¹. Ce moteur utilise à la fois le CPU et le GPU pour générer le rendu final. Dans un premier temps, une passe de lancer de rayons est calculée sur le CPU pour générer les informations nécessaires au calcul du descripteur : textures de normales, positions et profondeurs. Une autre texture est ensuite créée, coté GPU, pour y stocker le tenseur de courbure, extrait à partir des données géométriques. Ce tenseur est enfin utilisé par le moteur d'éclairage global pour modifier chacun des rayons lumineux par pixel.

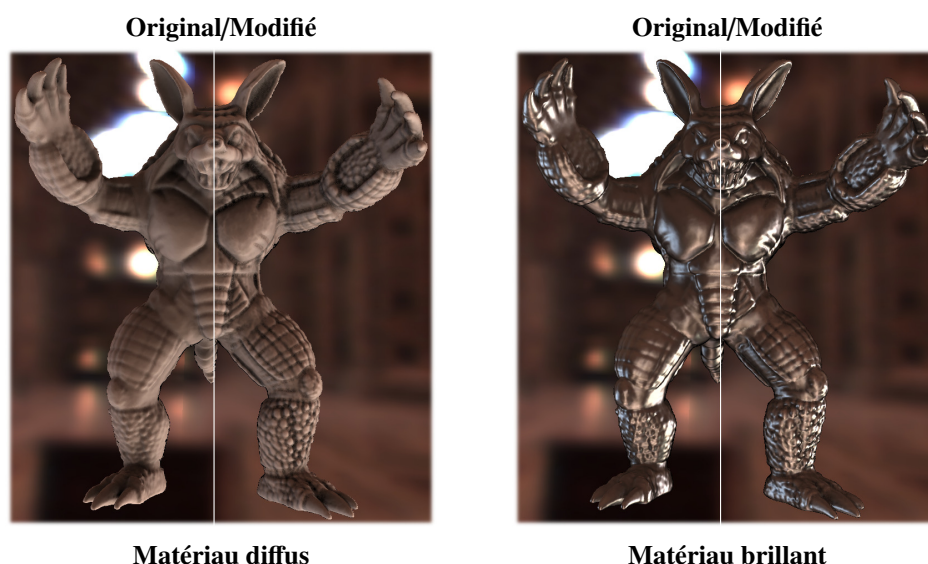


Figure 5.10: Le modèle Armadillo (©AIM@SHAPE) rendu avec un matériau diffus (à gauche) et légèrement spéculaire (à droite). Chacun des côtés montre le rendu original et modifié. Quel que soit le matériau, la mise en valeur des détails reste cohérente.

Des résultats sont présentés dans la Figure 5.10 avec deux types de matériaux différents. On y distingue très nettement les détails qui sont mis en valeur de manière cohérente après application de notre méthode, que le matériau soit diffus ou brillant. On remarque néanmoins que les effets sont un peu moins visibles lorsque l'objet devient spéculaire. Nous reviendrons sur cet effet à la fin du chapitre.

¹Ce moteur a été implémenté par Romain Pacanowski, co-auteur de l'article [VPB⁺09].

En modifiant l'éclairage incident de cette manière, nous changeons aussi la visibilité en chaque point de la surface. Cela améliore aussi la perception de la forme, car une partie plus ou moins large de l'environnement est considérée suivant que l'on applique une contraction ou une dilatation. Nous le démontrons plus précisément sur les rayons lancés pour calculer l'*Ambient Occlusion* (la visibilité moyenne des points de la surface par rapport à l'environnement). Nous modifions l'équation de l'AO définie dans le Chapitre 3 pour y introduire le terme permettant de modifier les rayons :

$$A'(\mathbf{p}) = \frac{1}{\pi} \int_{\Omega_n} \max(\mathbf{n} \cdot W(\boldsymbol{\ell}), 0) V(W(\boldsymbol{\ell})) d\boldsymbol{\ell}$$

La Figure 5.11 montre un exemple de calcul de l'AO modifié par notre méthode. On distingue les nombreux détails qui sont mis en valeur après application de notre méthode.



F . 5.11: À gauche : *Ambient Occlusion* classique. À droite : les rayons lancés sont modifiés avec notre technique pour révéler et exagérer de nombreux détails.

Style réaliste temps-réel

Le second moteur dans lequel cette méthode a été implémentée est un système de rendu entièrement sur le GPU. La méthode mise en œuvre est très simple et consiste à pré-échantillonner une carte d'environnement (typiquement entre 16 et 256 échantillons), puis à envoyer ces échantillons lumineux à la carte graphique pour évaluer l'équation. Un problème se pose dans ce cas pour activer la modification des rayons avec la formule actuelle : les rayons étant déjà échantillonnés, il n'est pas possible de connaître la valeur de la radiance entrante après modification des rayons car cette information est inaccessible. Pour contourner le problème et garder la possibilité de pré-échantillonner l'environnement, nous effectuons un

changement de variable dans l'équation modifiée 5.2 pour utiliser directement les directions modifiées :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_n} \rho(\mathbf{e}, W^{-1}(\ell')) (\mathbf{n} \cdot W^{-1}(\ell')) L(\mathbf{p} \leftarrow \ell') J d\ell'$$

On considère donc que les échantillons de lumière sont déjà modifiés dans cette équation, puis on utilise la fonction inverse W^{-1} pour retrouver les directions de lumières incidentes et calculer notamment la *BRDF* et l'orientation de la surface. Le jacobien J est introduit pour compenser la perte ou le gain d'énergie dû à la modification des rayons, et dépend naturellement de la fonction W :

$$J = \frac{4\lambda_u^3 \lambda_v^3 (1 + \ell'_z)^2}{(\lambda_u^2 \lambda_v^2 (1 + \ell'_z)^2 + \lambda_v^2 \ell'_u{}^2 + \lambda_u^2 \ell'_v{}^2)^2}$$

La dérivation complète de ce terme peut être trouvée en annexe A. C'est à partir de cette nouvelle formulation de l'équation de réflectance que les images de la Figure 5.12 ont été générées. Deux environnements différents ont été utilisés pour montrer la flexibilité de la méthode. Dans les deux cas, de nombreux détails sont mis en valeur, comme dans les cheveux ou sur les contours des yeux de l'enfant.

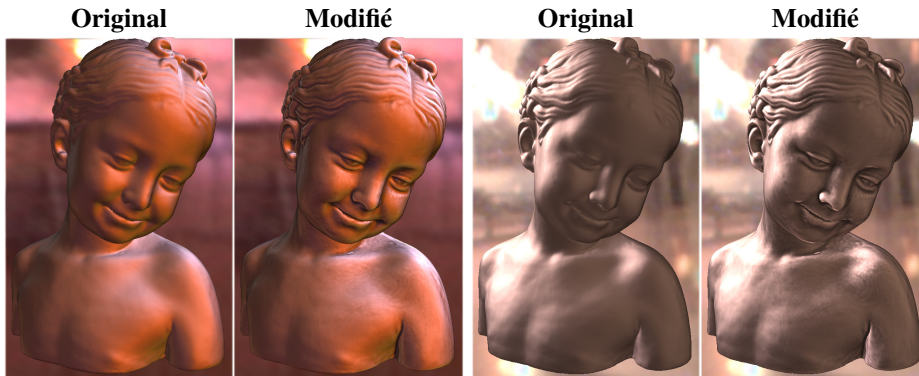


Fig. 5.12: Le modèle “Bambino”, rendu avec deux environnements lumineux différents. Ces résultats montrent les rendus originaux et ceux obtenus avec notre méthode. On distingue de nombreux détails qui apparaissent et mettent la forme en valeur de manière cohérente. Ces images ont été générées dans une fenêtre de 800×600 , avec 54 échantillons de lumière. La performance est de 37 *fps*.

Style cartoon

Cette méthode fonctionne aussi avec des styles moins réalistes. Nous utilisons ici la même méthode de quantification des couleurs [WOG06] que celle présentée dans la section précédente pour créer un effet cartoon dans l'image finale. La

préservation des matériaux ne nous semblant pas nécessaire pour un rendu de ce type, nous avons aussi ajouté un terme global dans l'équation du rendu, comme dans la section précédente, pour exagérer l'effet produit :

$$L'_\gamma(\mathbf{p} \rightarrow \mathbf{e}) = (\lambda_u \lambda_v)^\gamma L'(\mathbf{p} \rightarrow \mathbf{e}) \quad (5.3)$$

où $\lambda_{uv} \in [-1, 1]$ permet aussi de contrôler le contraste. Lorsque $\kappa_1 = 0$ et $\kappa_2 = 0$, alors $L'_\gamma = L'$. Dans les autres cas, le contraste augmente en fonction de la courbure et de l'intensité de modification des rayons. Deux exemples sont donnés dans la Figure 5.13. On y remarque que la technique permet aussi de mettre les détails de surface en valeur, avec des environnements naturels ou minimaux et des variations de couleurs simples.



FIG. 5.13: À gauche : un environnement naturel est utilisé pour rendre l'objet avant de quantifier les couleurs. La texture de la surface et les cavités sont mises en valeur après application de notre méthode. À droite : même avec un environnement minimaliste, composé d'un simple gradient du noir au blanc ici, notre méthode permet de révéler les détails de surface.

5.2.4 Comparaison avec les méthodes existantes

La technique que nous venons d'introduire, basée sur la manière dont notre système visuel perçoit la forme, fonctionne avec des environnements, des matériaux et des styles arbitraires. Les résultats produits peuvent par exemple être comparés aux techniques d'exagération des détails de Rusinkiewicz *et al.* [RBD06] ou de Ritschel *et al.* [RSI⁺08] (cf. Figure 5.14). Comparé à [RBD06], nous pouvons mettre en valeur les détails de surface avec une variété de matériaux beaucoup plus grande. De plus, notre technique ne nécessite pas de précalculs et reste cohérente spatialement et temporellement. L'utilisation de notre descripteur dépendant du point de vue permet aussi d'avoir des niveaux de détail automatiques et contrôlables. Par rapport à [RSI⁺08] qui augmente le contraste de la radiance sans discrimination et modifie la perception des matériaux, notre système offre un meilleur contrôle car il

permet de mettre en valeur les détails de manière uniforme. Nous proposons aussi une technique très simple, avec seulement trois paramètres utilisateur : l'intensité de courbure α , le contraste λ et le paramètre s qui permet de sélectionner le niveau de détail dans le descripteur.

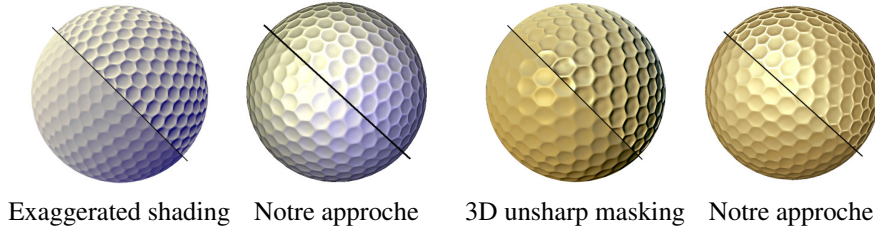


FIG. 5.14: Contrairement aux autres approches, notre technique permet de modifier les détails de surface avec des matériaux qui ne sont pas forcément diffus. À gauche : les travaux présentés dans [RBD06] sont limités à un *cosine shading*. À droite : la perception des matériaux est altérée dans [RSI⁺08], et l'exagération des détails n'est pas uniforme.

5.3 Modification des lumières réfléchies

Comme nous venons de le voir, le style obtenu par la modification des rayons lumineux est contraint par le fait que nous devons disposer d'un certain nombre d'échantillons lumineux. C'est la raison pour laquelle nous avons mis en place une troisième méthode, plus flexible, pour contrôler la manière dont les détails sont mis en valeur. L'observation clé est que les variations des motifs lumineux créés avec la méthode précédente sont très corrélées aux types des surfaces mises en valeur. En regardant par exemple l'objet diffus de la Figure 5.10, nous pouvons remarquer que l'effet obtenu est l'assombrissement des concavités et l'éclaircissement des convexités. L'idée principale de notre nouvelle méthode, que nous avons appelée *Radiance Scaling* (RS), est d'ajuster l'intensité réfléchie de chacune des lumières en prenant en compte la courbure de la surface, mais aussi le matériau utilisé (Figure 5.2 (c)). RS a été accepté à la conférence I3D [VPB⁺10] et va paraître sous forme étendue dans le journal TVCG. Formellement, on réécrit l'équation de réflexion de la manière suivante :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_n} \rho(\mathbf{e}, \boldsymbol{\ell})(\mathbf{n} \cdot \boldsymbol{\ell}) \sigma(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell}) L(\mathbf{p} \leftarrow \boldsymbol{\ell}) d\boldsymbol{\ell} \quad (5.4)$$

où σ est la fonction de modification de la radiance. Intuitivement, il s'agit d'augmenter ou de diminuer localement l'intensité de chacune des sources lumineuses pour accentuer les variations de courbure dans le rendu final. Prenons l'exemple d'un reflet brillant à la surface d'un objet : si l'on accentue l'intensité lumineuse sur les convexités et qu'on la diminue sur les concavités, l'effet obtenu sera d'allonger le reflet le long de la convexité, et permettra ainsi de mieux distinguer les

détails de surface. Cet ajustement prend en compte la courbure de la surface, mais aussi les paramètres de la *BRDF* pour éviter de briser trop fortement la perception des matériaux. La fonction de mise à l'échelle est donc une notation simplifiée pour $\sigma_{\alpha,\gamma}(\kappa(\mathbf{p}), \delta(\mathbf{e}, \boldsymbol{\ell}))$.

- La fonction de mappage de courbure $\kappa(\mathbf{p}) : \mathbb{R}^3 \rightarrow [-1, 1]$ permet de normaliser les intensités de courbures, où -1 correspond à la concavité la plus élevée, 0 aux régions planaires et 1 à la convexité maximum.
- La fonction de mappage de la réflectance $\delta(\mathbf{e}, \boldsymbol{\ell}) : \Omega^2 \rightarrow [0, 1]$ calcule des valeurs normalisées où 0 et 1 correspondent respectivement aux intensités minimum et maximum. Intuitivement, cette valeur permet d'identifier les directions de lumière qui contribuent le plus à l'intensité réfléchie.

Une telle fonction procure de nombreux avantages sur le contrôle du message et la variété des styles pouvant être créés. Si l'on s'éloigne des intuitions exprimées dans [FTA04, FTA09], on gagne en flexibilité et en rapidité. Par exemple, il est possible de contrôler séparément les différents composants (diffus/spéculaire/ambiant) de la *BRDF* et donc, de préciser les lieux où l'exagération aura lieu. Le fait de rajouter une simple fonction multiplicative dans l'équation du rendu permet d'intégrer cette technique dans de nombreux algorithmes existants. Nous montrerons les différentes applications dans la suite de cette section, après avoir présenté la fonction de mise à l'échelle.

5.3.1 Fonction de mise à l'échelle

Le message que nous souhaitons communiquer avec cette méthode consiste à corréliser les concavités et convexités à l'augmentation ou à la diminution de l'intensité lumineuse. La fonction de mise à l'échelle est conçue de telle sorte qu'elle exagère les détails de surface en prenant en compte trois propriétés :

1. La fonction doit être monotone pour qu'aucun nouvel extrema ne soit ajouté dans le rendu.
2. La fonction ne doit pas avoir d'influence sur la radiance réfléchie lorsqu'aucun détail n'apparaît dans le tenseur (i.e. lorsque la surface est planaire). Dans ce cas, $\sigma = 1$.
3. La manière dont la fonction éclaircit ($\sigma < 1$) ou assombrit ($\sigma > 1$) les détails doit pouvoir se contrôler facilement avec un simple paramètre et doit avoir des effets symétriques autour du point d'invariance.

La fonction rationnelle linéaire suivante satisfait ces propriétés :

$$\sigma_{\alpha,\gamma}(\kappa, \delta) = \frac{\alpha e^{\gamma\kappa} + \delta(1 - \alpha(1 + e^{\gamma\kappa}))}{\alpha + \delta(e^{\gamma\kappa} - \alpha(1 + e^{\gamma\kappa}))} \quad (5.5)$$

où $\alpha \in]0, 1[$ contrôle la position du point d'invariance de la fonction σ et $\gamma \in [0, \infty[$ est l'intensité avec laquelle la mise à l'échelle est appliquée. Le point d'invariance

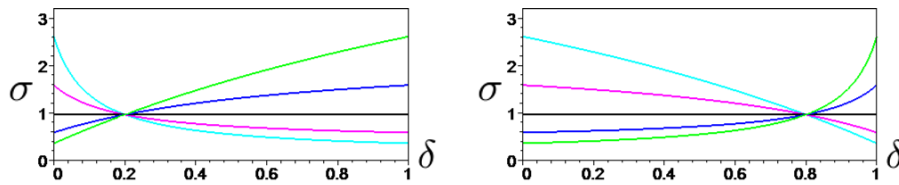


Fig. 5.15: Deux graphes de la fonction de mise à l'échelle σ avec différentes valeurs pour le point d'invariance (à gauche : $\alpha = 0.2$; à droite : $\alpha = 0.8$), et avec des intensités de courbures croissantes $\kappa = \{-1, -1/2, 0, 1/2, 1\}$.

permet de contrôler comment les variations dans le rendu communiquent les informations sur la forme. Sur les convexités, les intensités de lumière reflétées dont la valeur obtenue est au-dessus de α sont éclaircies tandis que les valeurs qui sont en dessous sont assombries. L'effet opposé est obtenu pour les concavités. La Figure 5.16 illustre plusieurs valeurs de α . Comme on peut le constater sur les graphes de la Figure 5.15, la fonction σ possède des propriétés intéressantes qui satisfont les contraintes initiales. On peut noter, par exemple, que la fonction est égale à 1 seulement lorsque la fonction de mappage des réflectances $\delta = \alpha$ ou lorsque la courbure $\kappa = 0$. Ensuite, les détails convexes et concaves ont un effet réciproque sur la fonction : $\sigma_{\alpha,\gamma}(\kappa, \delta) = 1/\sigma_{\alpha,\gamma}(-\kappa, \delta)$. Enfin, une troisième propriété est que la fonction est symétrique par rapport à α : $\sigma_{\alpha,\gamma}(\kappa, 1 - \delta) = 1/\sigma_{1-\alpha,\gamma}(\kappa, \delta)$. Ces choix font que la fonction de mise à l'échelle est facile à manipuler pour les utilisateurs (cf. Figure 5.16).

Il nous faut maintenant définir comment sont calculées les fonctions de mappage de courbure et de réflectance. Comme dans la section précédente, nous fournissons une fonction de mappage de courbure anisotrope, qui dépend de chacune des di-

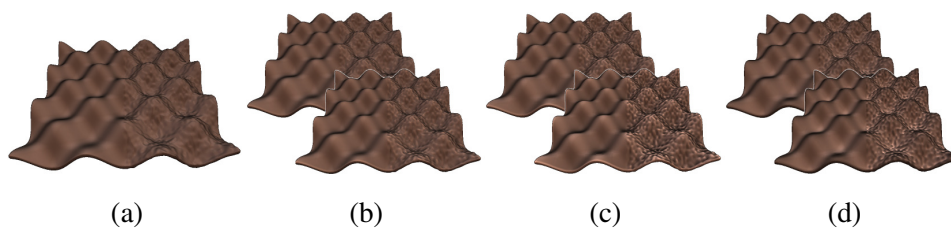


Fig. 5.16: Effet du paramètre de mise à l'échelle. (a) Rendu original, avec $\gamma = 0$. (b-d) La forme est mise en valeur avec deux coefficients de mise à l'échelle (en haut : $\gamma = 10$; en bas : $\gamma = 20$) et trois points d'invariance : (b) $\alpha = 0.5$ donne un bon équilibre entre la mise en valeur des concavités et des convexités ; (c) $\alpha = 0.01$, les convexités sont éclaircies ; (d) $\alpha = 0.99$, les variations du rendu sont inversées.



Fig. 5.17: Effet de l'anisotropie sur un objet éclairé par une lumière directionnelle venant de la gauche. Sur l'exemple de gauche : l'anisotropie est orthogonale à la direction de lumière ($\lambda = -1$). Au centre : la mise en valeur est isotrope ($\lambda = 0$). A droite : l'anisotropie a lieu dans la même direction de la lumière ($\lambda = 1$). Dans chacun des cas, ce sont des détails différents qui sont mis en valeur.

directions lumineuses ℓ :

$$\kappa(\mathbf{p}, \ell) = \tanh\left((H + \lambda\Delta_\kappa)\ell_u^2 + (H - \lambda\Delta_\kappa)\ell_v^2 + H\ell_z^2\right)$$

La direction de lumière $\ell = (\ell_u, \ell_v, \ell_z)$ est exprimée dans le repère $(\mathbf{u}, \mathbf{v}, \mathbf{z})$, où \mathbf{u} et \mathbf{v} représentent les directions de courbures principales. H est, comme on l'a déjà vu, la courbure moyenne et $\Delta_\kappa = \kappa_1 - \kappa_2$ est une mesure de l'anisotropie de courbure. Intuitivement, la fonction renvoie une valeur de courbure obtenue par un mélange linéaire entre les courbures principales et la courbure moyenne en fonction de la projection de ℓ sur le plan de l'image. Le paramètre $\lambda \in [-1, 1]$ permet à l'utilisateur de contrôler la manière dont l'anisotropie est prise en compte (cf. Figure 5.17) : lorsque $\lambda = 0$, l'effet est isotrope ($\forall \ell, \kappa(\mathbf{p}, \ell) = \tanh(H)$); si $\lambda = 1$, la mise en valeur est anisotrope ($\kappa(\mathbf{p}, \mathbf{u}) = \tanh(\kappa_1)$); si enfin $\lambda = -1$, l'effet est anisotrope mais dans la direction opposée ($\kappa(\mathbf{p}, \mathbf{u}) = \tanh(\kappa_2)$). Dans le cas où la direction lumineuse ℓ est alignée avec \mathbf{z} , seule la mesure isotrope est prise en compte.

5.3.2 Styles

La fonction de mappage de l'intensité reflétée δ dépend de la *BRDF* utilisée. Nous montrons son intégration dans une grande variété de scénarios pour démontrer sa facilité d'utilisation et sa flexibilité.

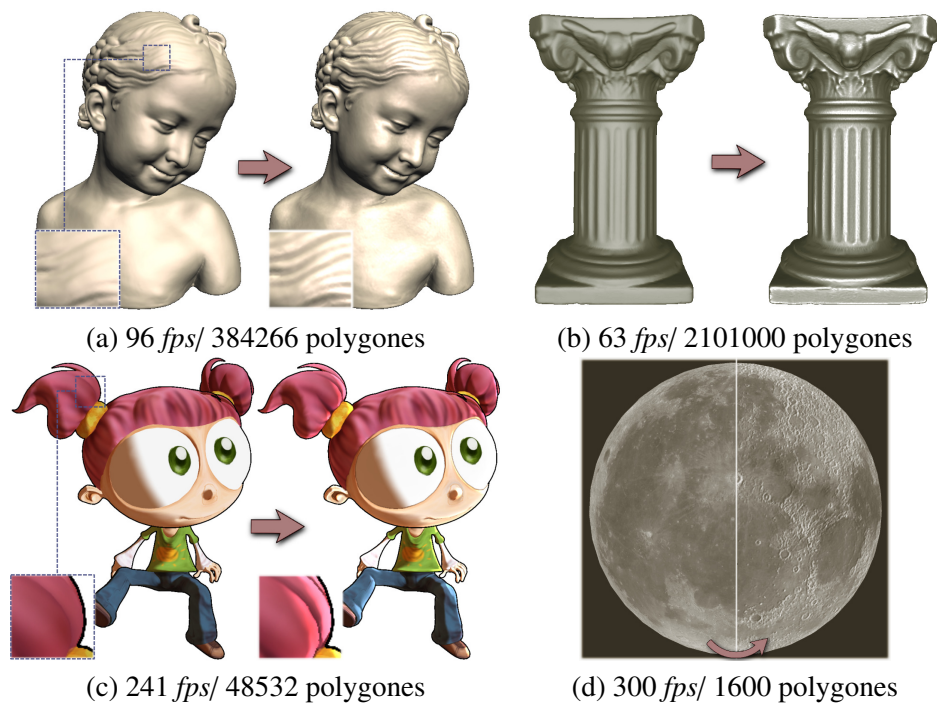
Eclairage de Phong simple

Des modèles simples, avec peu de sources lumineuses, sont généralement utilisés dans les applications interactives comme les jeux vidéo. Le modèle de Phong

[BT75] introduit par l'équation 3.3 du Chapitre 3 est l'un d'entre eux. Notre fonction de mise à l'échelle de la radiance s'y intègre très facilement et permet aussi de contrôler séparément les différents composants de la *BRDF*. Avec une seule lumière, l'équation de Phong avec *Radiance Scaling* se réécrit de la manière suivante :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \sum_j \rho_j(\mathbf{e}, \boldsymbol{\ell}_0) \sigma_j(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell}_0) L_j(\boldsymbol{\ell}_0)$$

où $j \in \{a, d, s\}$ itèrent sur les composants ambiants, diffus et spéculaires du modèle d'ombrage. La différence principale entre les termes de rendu réside dans le choix de la fonction de mappage d'intensité. Comme les lobes de Phong sont déjà définis dans l'intervalle $[0, 1]$, le choix le plus naturel est de les utiliser directement



F . 5.18: La technique illustrée dans des scénarios de rendus simples. (a) Chaque lobe du modèle de Phong est modifié indépendamment pour révéler les détails de surface, comme dans les cheveux ici. (b) Cette méthode est équivalente au *Mean Curvature Shading* lorsqu'elle est appliquée seulement au lobe ambiant (nous l'avons aussi combiné au terme diffus dans cette image). (c) Les détails de surface sont mis en valeur quel que soit le style ; on distingue beaucoup mieux les détails des cheveux, de la jambe droite ou autour du nez avec ce rendu cartoon. (d) De nombreux détails apparaissent sur cette sphère combinée à une carte de normales et composée d'un matériau sub-Lambertien.

pour la fonction de mappage des intensités : $\delta_j = \rho_j$. Cela permet non seulement d'identifier les directions de référence dans lesquelles l'intensité de réflexion de la lumière est maximale (i.e. \mathbf{n} pour δ_d ou \mathbf{r} pour δ_s), mais fournit aussi une décroissance non-linéaire lorsqu'on s'éloigne de cette direction. Chacun des termes peut être mis en valeur indépendamment avec des paramètres différents : γ_a , γ_d et γ_s .

La Figure 5.18 montre des résultats obtenus avec le modèle de Phong modifié, avec une simple lumière directionnelle. L'effet de l'anisotropie étant très sensible avec un tel éclairage, nous utilisons des valeurs de λ assez faibles pour ces configurations. On remarque notamment que l'effet de notre méthode sur le seul terme ambiant du modèle de Phong donne des résultats équivalents au *Mean Curvature Shading* [KWTM03] (Figure 5.18 (b)). Sur un rendu de type cartoon (c), où les couleurs ont été quantifiées, la fonction σ permet de préciser la forme des objets. Nous avons aussi appliqué notre méthode sur des objets composés de matériaux sub-Lambertiens (d) ($\rho_{sl}(\boldsymbol{\ell}_0) = (\mathbf{n} \cdot \boldsymbol{\ell}_0)^\zeta$, $\zeta \in [0, 1[$, avec $\delta_{sl} = \rho_{sl}$). La Lune est ici illustrée ($\zeta = 0.5$) et modélisée avec une simple sphère et une carte de normales détaillée.

Comme on peut le voir de part les performances obtenues, cette fonction peut être facilement intégrée dans les applications nécessitant des contraintes temps-réel comme les jeux vidéo. Nous avons implémenté une version optimisée de cette méthode (avec une seule lumière directionnelle et sans diffusion pour le descripteur) et nous avons mesuré un surplus de 0.17 milliseconde par image dans une fenêtre de 1024×768 . À noter que ces performances ne dépendent pas de la complexité de la scène car tout est calculé en espace image.

Eclairage complexe

La fonction de mise à l'échelle peut être implémentée avec des environnements plus complexes et des modèles de matériaux plus précis. Pour la tester, nous l'avons intégrée avec un environnement composé de multiples échantillons lumineux et avec la *BRDF* d'Ashikhmin *et al.* [APS00]. Ici aussi, nous laissons la possibilité à l'utilisateur de contrôler séparément les composants de la *BRDF*. Avec N sources lumineuses, la version discrétisée de l'équation 3.1 devient :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \sum_{i=1}^N \rho_d(\boldsymbol{\ell}_i) \sigma_d(\mathbf{p}, \boldsymbol{\ell}_i) L(\boldsymbol{\ell}_i) + \sum_{i=1}^N \rho_s(\mathbf{e}, \boldsymbol{\ell}_i) \sigma_s(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell}_i) L(\boldsymbol{\ell}_i)$$

où $\boldsymbol{\ell}_i$ représente la i ème direction de lumière au point \mathbf{p} et ρ_d , ρ_s correspondent respectivement aux lobes diffus et spéculaires de la *BRDF*. Contrairement au modèle de Phong, les lobes de la *BRDF* d'Ashikhmin *et al.* [AS00] peuvent se situer en dehors de l'intervalle $[0, 1]$ et ne peuvent donc pas être utilisés directement pour la fonction δ . L'alternative que nous proposons est donc de se baser sur les directions de référence de chacun des lobes pour calculer la fonction de mappage de réflectance : $\delta_d(\boldsymbol{\ell}_i) = (\boldsymbol{\ell}_i \cdot \mathbf{n})$ pour le terme diffus et $\delta_s(\mathbf{e}, \boldsymbol{\ell}_i) = (\mathbf{h}_i \cdot \mathbf{n})$

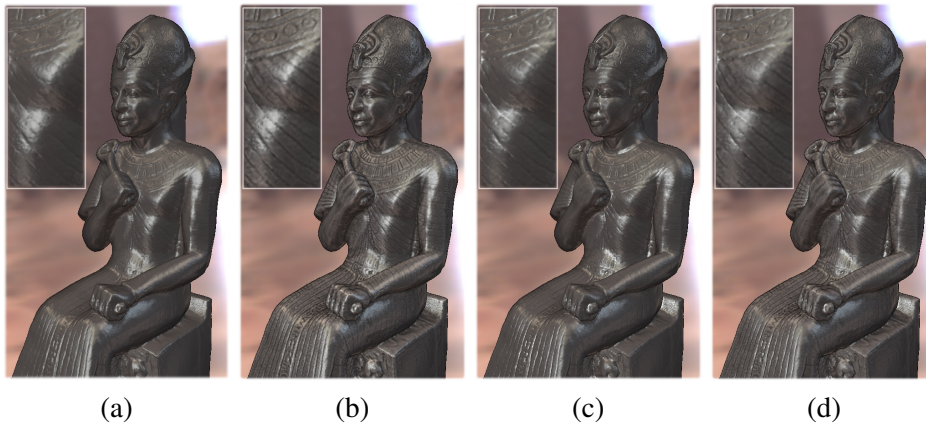


Fig. 5.19: (a) Rendu original d'un objet composé d'un matériau brillant avec la *BRDF* d'Ashikmin *et al.* [APS00], avec un zoom sur le torse. (b) Appliquer la méthode sur le seul terme diffus a pour effet de mettre en valeur les surfaces qui sont éloignées des tâches lumineuses en assombrissant les rainures se trouvant sur le torse ou sur les bras. (c) En l'appliquant seulement sur le composant spéculaire, des effets différents apparaissent : certaines rainures sont éclaircies et les régions rasantes (par rapport au point de vue) sont mises en valeur. (d) les deux résultats précédents sont combinés pour obtenir un résultat contenant des informations très riches sur la forme. Des alternances subtiles de zones claires et sombres sur le torse montrent de nombreux détails.

pour le terme spéculaire, \mathbf{h}_i étant le demi-vecteur entre la direction ℓ_i et la vue \mathbf{e} . Comme précédemment, chacun des termes peut être contrôlé séparément avec les paramètres γ_d et γ_s . La Figure 5.19 (a) montre un exemple de rendu temps-réel obtenu avec un objet brillant, la *BRDF* d'Ashikmin et des échantillons lumineux prélevés sur une carte d'environnement comme dans la section précédente. Le composant diffus est tout d'abord mis en valeur (Figure 5.19 (b)), ce qui a pour effet d'assombrir les concavités et d'éclaircir les convexités. On peut le voir par exemple sur le torse, les bras, la tunique, ou encore la coiffe de Ramsès. Son visage donne aussi une bonne illustration de la manière dont les variations sont introduites dans le rendu : la forme des yeux, de la bouche, ou les rides du front apparaissent plus clairement car les concavités et convexités proches ont pour effet de contraster le gradient du terme diffus. Lorsque seul le terme spéculaire est mis en valeur (Figure 5.19 (c)), les détails sur la tunique de la statue sont mis en valeur et d'autres apparaissent sur le torse et la coiffe. La dernière image (Figure 5.19 (d)) montre le résultat obtenu en combinant les deux effets précédents. Les détails sont mis en avant différemment suivant qu'ils se trouvent dans une région concernée par le terme diffus ou spéculaire de la *BRDF*, tout en conservant une apparence globalement cohérente.

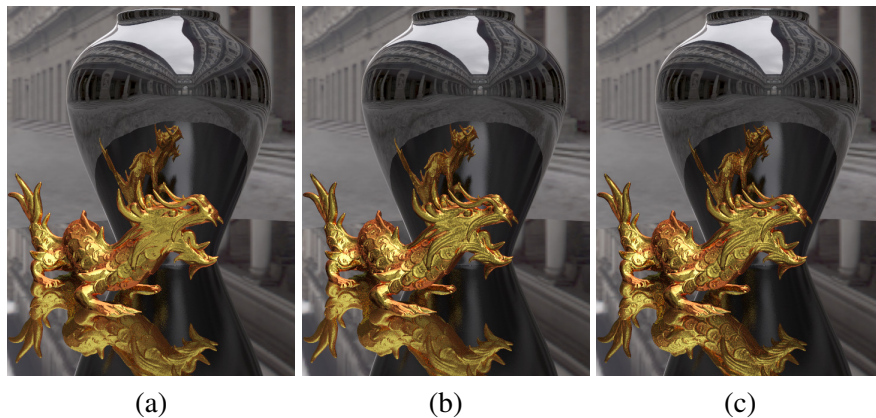


FIG. 5.20: Modification de la radiance avec inter-réflexions. (a) Une scène 3D, composée d'un modèle de dragon avec un matériau brillant et d'un vase posé sur un plan, tous deux avec un matériau miroir. (b) Notre méthode, appliquée sur le premier rebond, met les détails de surface du dragon en valeur sans modifier les réflexions de celui-ci. (c) Appliquée sur tous les rebonds des rayons, les détails du dragon apparaissent sur sa réflexion dans la table et sur le vase.

Nous avons aussi implémenté cet algorithme dans un moteur d'éclairage global pour ajouter des effets tels que les inter-réflexions et les ombres douces. La méthode s'intègre exactement de la même manière que précédemment, en ajoutant la fonction σ directement dans l'équation originale. La Figure 5.20 montre une scène composée d'objets plus ou moins réfléchissants. Elle montre notamment qu'il est possible de choisir les rayons sur lesquels la méthode est appliquée. Seul le premier rebond est mis en valeur dans la figure du milieu tandis la fonction est appliquée sur tous les rebonds sur l'image de droite ; apparaissent alors les détails dans les réflexions du dragon. À noter que dans cet exemple, c'est le descripteur calculé en espace objet [Rus04] qui a été utilisé pour des raisons pratiques. En testant avec d'autres exemples, nous avons aussi remarqué qu'elle fonctionnait bien sur des scènes contenant des ombres douces (cf. Figure 5.21). Dans ce cas, la forme des ombres n'est pas altérée, mais les détails de surface sont toujours mis en valeur. Le temps de rendu pour ces algorithmes d'éclairage global est négligeable lorsqu'on intègre la fonction de mise à l'échelle. La différence est inférieure à une seconde entre un rendu original et un rendu avec intégration de notre méthode, alors que le temps de rendu d'une image est d'environ dix minutes.

Environnements précalculés

Les algorithmes d'éclairage global sont généralement des processus gourmands en espace mémoire et en temps de calcul. Pour ces raisons, de nombreuses

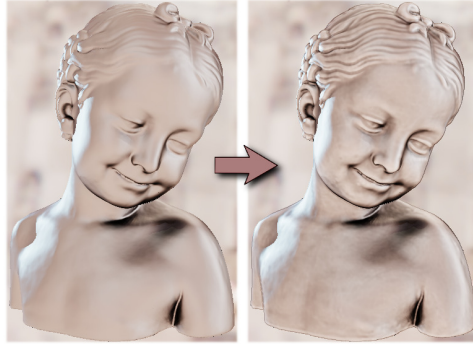


FIG. 5.21: Dans cet exemple, la visibilité est calculée par rayon lumineux et engendre des ombres douces (à gauche). En appliquant notre technique dans cette configuration, *Radiancance Scaling* conserve la forme générale des ombres tout en exagérant les détails de surface de manière appropriée.

méthodes ont été proposées pour précalculer et réutiliser les données de radiance. Dans le cas général, notre technique consiste à introduire un terme additionnel σ dans l'équation du rendu 3.1, et dépend à la fois d'une fonction de courbure $\kappa(\mathbf{p})$ et d'une fonction de réflectance $\delta(\mathbf{e}, \boldsymbol{\ell})$; ce qui correspond à, au moins, une dimension supplémentaire dans le cas du précalcul de radiance déjà mise en valeur. Pour éviter d'avoir à stocker une dimension supplémentaire, nous remplaçons la fonction de mappage de réflectance $\delta(\mathbf{e}, \boldsymbol{\ell})$ par une version simplifiée $\bar{\delta}(\mathbf{e})$, indépendante de la direction lumineuse $\boldsymbol{\ell}$. La fonction de mise à l'échelle $\sigma_{\alpha,\gamma}(\kappa(\mathbf{p}), \delta(\mathbf{e}, \boldsymbol{\ell}))$ est donc aussi remplacée par une version simplifiée $\bar{\sigma}_{\alpha,\gamma}(\kappa(\mathbf{p}), \bar{\delta}(\mathbf{e}))$, notée $\bar{\sigma}(\mathbf{p}, \mathbf{e})$ et sortie de l'intégrale dans l'équation 5.4 :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \bar{\sigma}(\mathbf{p}, \mathbf{e}) \int_{\Omega_n} \rho(\mathbf{e}, \boldsymbol{\ell}) (\mathbf{n} \cdot \boldsymbol{\ell}) L(\mathbf{p} \leftarrow \boldsymbol{\ell}) d\boldsymbol{\ell}. \quad (5.6)$$

Il est important de noter que cette version se rapproche de celle qui a été introduite dans la première partie de ce chapitre, où l'on modifiait chacune des couleurs du rendu en fonction d'une valeur de courbure. La différence ici est que nous disposons toujours d'une information directionnelle pour la fonction σ . Comme elle n'est plus appliquée par source lumineuse, il faut définir de nouvelles fonctions de réflectances $\bar{\delta}_d$ et $\bar{\delta}_s$ pour les termes diffus et spéculaires.

Concernant les matériaux diffus, les techniques d'*Ambient Occlusion* [PG04] et les cartes d'environnement pré-filtrées [KVHS00] sont largement utilisées (et combinées) pour calculer la radiance. Une approximation similaire peut être utilisée en intégrant notre fonction de mise à l'échelle; si l'on considère une *BRDF* constante ($\rho(\mathbf{e}, \boldsymbol{\ell}) = \rho_d$) et un éclairage direct fait depuis une carte d'environnement ($L(\mathbf{p} \leftarrow \boldsymbol{\ell}) = V(\boldsymbol{\ell})L_{env}(\boldsymbol{\ell})$ où $V \in \{0, 1\}$) est le terme de visibilité et L_{env} est la carte

d'environnement), alors l'équation 5.6 devient :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \bar{\sigma}(\mathbf{p}, \mathbf{e}) \rho_d \int_{\Omega_n} (\mathbf{n} \cdot \boldsymbol{\ell}) V(\boldsymbol{\ell}) L_{env}(\boldsymbol{\ell}) d\boldsymbol{\ell}.$$

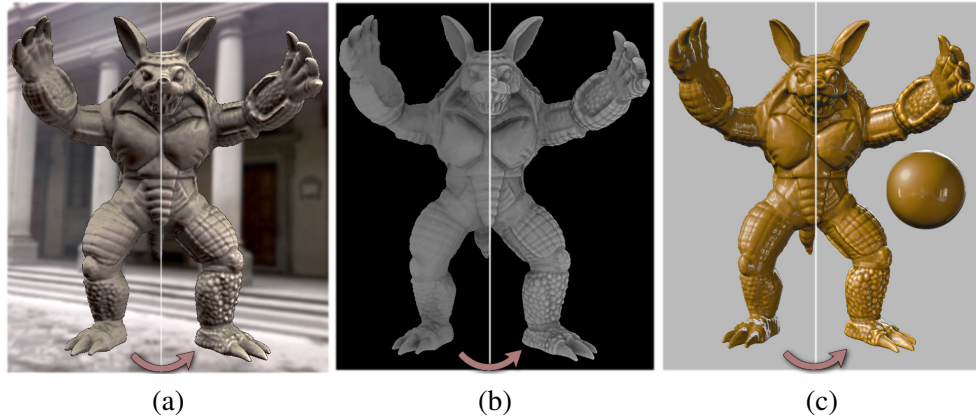
On peut donc approcher la radiance modifiée avec l'équation suivante :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) \simeq \bar{\sigma}(\mathbf{p}, \mathbf{e}) \rho_d A(p) \bar{L}(\mathbf{n})$$

$A(p)$ étant l'AO stockée par sommet et \bar{L} est une moyenne d'irradiance, stockée dans une carte d'environnement :

$$\bar{L}(\mathbf{n}) = \int_{\Omega_n} L_{env}(\boldsymbol{\ell}) d\boldsymbol{\ell}$$

Dans le cas des matériaux diffus, nous utilisons la fonction de mappage de réflectance $\bar{\delta}_d(\mathbf{p}) = \bar{L}(\mathbf{n})/\bar{L}^*$, \mathbf{n} étant la normale au point \mathbf{p} , et $\bar{L}^* = \max_{\mathbf{n}} \bar{L}(\mathbf{n})$ la radiance moyenne trouvée dans la carte d'environnement. Ce choix est cohérent avec les matériaux parfaitement diffus, car dans ce cas, la direction de lumière qui contribue le plus à l'intensité reflétée est en moyenne la direction de la normale. La



F . 5.22: (a) Résultat utilisant un environnement pré-filtré pour stocker l'irradiance et l'AO pour stocker la visibilité moyenne par sommet. Notre fonction s'adapte facilement à ce scénario et exagère les détails en temps-réel (66 *fps*/ 345944 polygones). (b) En l'appliquant seulement à l'AO, les résultats sont aussi convaincants. (c) Nous avons aussi testé la fonction avec des rendus stylisés. Notre fonction est ici appliquée aux *Lit-Sphere* [SMGG01].

Figure 5.22 (a) montre l'effet de notre méthode sur un objet éclairé par un environnement pré-filtré et combiné avec l'AO pour représenter le terme de visibilité. On peut observer comment les motifs géométriques sont mis en valeur sur la jambe, le bras ou le front. L'AO est montrée séparément dans la Figure 5.22 (b).

Une alternative consiste à utiliser des environnements pré-filtrés dans le but d'obtenir des styles de rendu variés (Figure 5.22 (c)). Les *Lit Sphere* [SMGG01] sont ici combinées à notre fonction de mise à l'échelle. Le principe est le même dans le cas des matériaux parfaitement réfléchissants ou réfractifs. Dans ce cas, la *BRDF* est une fonction de Dirac dans la direction de réflexion/réfraction, et on ignore le terme de visibilité :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \bar{\sigma}(\mathbf{p}, \mathbf{e})L_{env}(\mathbf{r})$$

La fonction de mappage de réflectance est ici égale à $\bar{\delta}_s(\mathbf{e}) = L_{env}(\mathbf{r})/L_{env}^*$, \mathbf{r} étant la direction de vue reflétée/réfractée et $L_{env}^* = \max_{\mathbf{r}} L_{env}(\mathbf{r})$ l'irradiance maximum dans la carte d'environnement. Une fois de plus, ce choix est cohérent avec ce type de matériaux, car dans ce cas, la direction de lumière qui contribue le plus à l'intensité obtenue est la direction de vue reflétée/réfractée. La Figure 5.23 montre

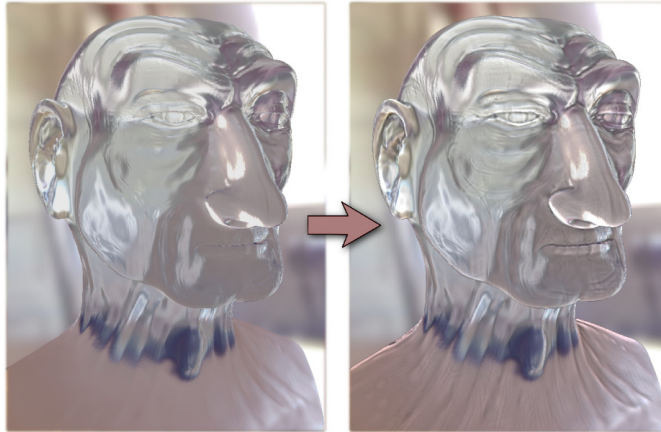
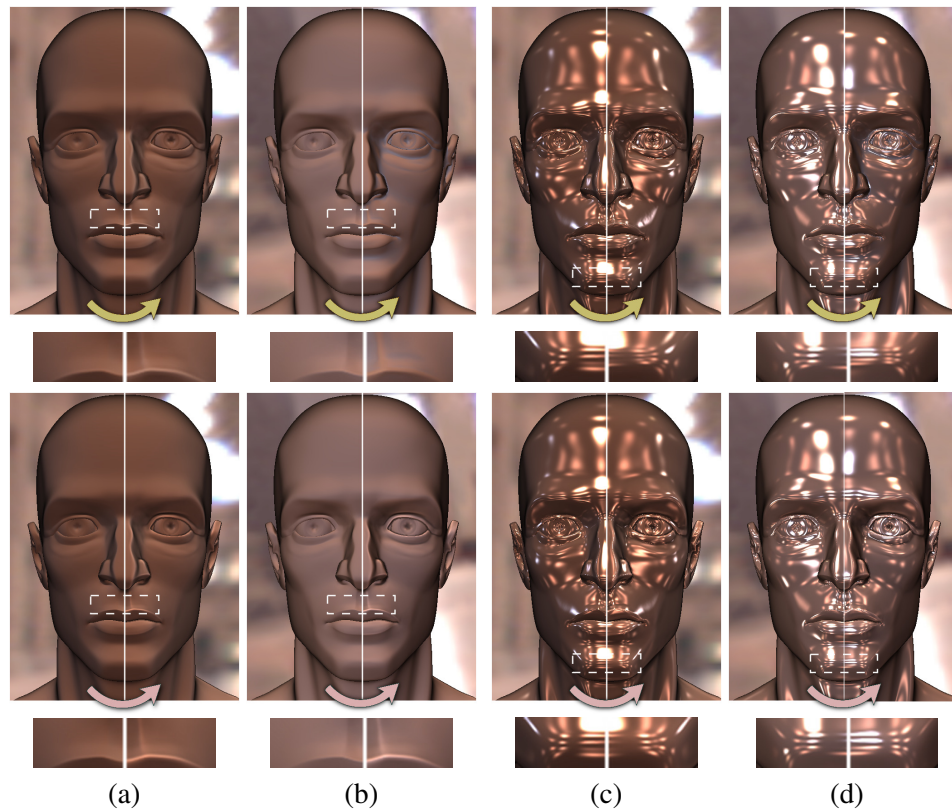


Figure 5.23: Même avec un matériau parfaitement réfractif (nous utilisons ici une simple approximation avec un seul rebond), notre méthode est capable de communiquer les détails de surface efficacement.

un exemple avec une simple approximation d'un matériau parfaitement réfractif. Notre approche permet aussi de mettre en valeur des surfaces miroir avec une approche très similaire, en utilisant le vecteur de vue reflété.

5.4 Comparaison des approches

Il est intéressant de comparer les effets de chacune de ces méthodes pour comprendre comment elles sont appliquées et pourquoi elles permettent de mettre en valeur la forme. Dans ce chapitre, nous avons vu trois méthodes pour exagérer les détails ; chacune d'entre elles est basée sur le même descripteur de forme dépendant du point de vue (cf. Chapitre 4), mais avec un message sensiblement différent, contraint par la manière d'intégrer la courbure dans l'équation du rendu et donc



F . 5.24: Comparaison entre les méthodes d'exagération vues dans les Sections 5.2 et 5.3 de ce chapitre. La technique de modification des rayons lumineux (LW) est illustrée dans la ligne du haut tandis que celle qui modifie les intensités (RS) se trouve dans la ligne du bas. (a-b) Les effets obtenus par les deux méthodes sont similaires quand elles sont appliquées sur des matériaux diffus : la plupart du temps, les convexités sont rendues plus claires que les concavités, qui sont assombries. Pour certaines orientations du point de vue par rapport à l'environnement, la technique de modification des rayons renverse cet effet (les concavités sont plus claires que les convexités). (c-d) Sur une surface brillante, on remarque que l'effet est de modifier la fréquence du signal (dans le cas où l'on modifie les rayons) et l'amplitude du signal (dans le cas de la mise à l'échelle de l'intensité).

par le style. Parmi les trois méthodes que nous avons introduites (cf. Figure 5.2), la première (*Apparent Relief* - AR) et la troisième (*Radiance Scaling* - RS) modifient l'intensité du rendu tandis que la seconde (*Light Warping* - LW) modifie les variations des motifs lumineux. Comme RS est plus flexible que AR (grâce à l'utilisation de matériaux et de styles plus variés) et que RS et LW ont des formulations très proches, nous comparons ces deux dernières techniques dans cette section.

La Figure 5.24 montre une comparaison des effets de ces deux méthodes avec le même style de rendu (réaliste dans ce cas) et un objet composé du même matériau. Ainsi, les différences montrent bien comment l'application de la courbure sur le rendu peut influencer le résultat. Deux matériaux différents sont comparés ici. Dans le cas d'un matériau diffus, on remarque que les deux méthodes ont des effets sensiblement similaires car elles tendent à assombrir les concavités et à éclaircir les convexités. Cependant, cet effet peut être inversé par LW pour certaines orientations de l'environnement. La principale différence entre les deux techniques apparaît dans le cas d'un matériau spéculaire. LW modifie la fréquence des motifs lumineux reflétés sur la surface, tandis que RS modifie l'amplitude du signal obtenu (cf. Figure 5.24).

Au niveau des performances, RS est plus rapide que LW (cf. Figure 5.25). On remarque que quelle que soit la méthode utilisée, le temps de rendu d'une image augmente linéairement avec le nombre d'échantillons de lumières pris en compte dans l'environnement. Cette différence est principalement due à la projection stéréographique et au calcul de la jacobienne dans l'équation inverse, qui ajoutent des opérations complexes dans les shaders.

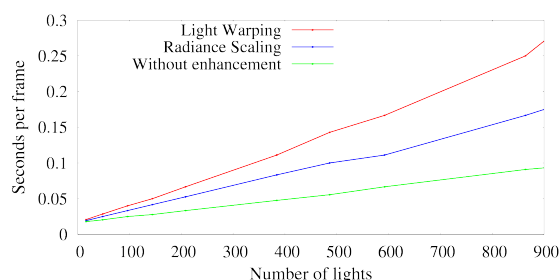


Figure 5.25: Ces courbes montrent les performances obtenues avec la scène de la Figure 5.19 sans exagération puis avec les deux méthodes de modification des intensités et de modification des rayons lumineux. Le modèle 3D est composé de 1652528 polygones. Dans tous les cas, le temps de rendu augmente linéairement avec le nombre de lumières utilisées. Mais la version de modification des intensités est aussi linéairement plus rapide que celle qui modifie les rayons.

Nous avons aussi essayé de combiner ces deux méthodes et avons trouvé que cette double exagération est efficace sur les objets brillants, comme on peut le voir sur la Figure 5.26. Dans ce cas, la technique de modification des rayons lumineux déforme les motifs lumineux de manière à différencier au mieux les convexités et concavités, en se basant sur un environnement contenant assez de variations. Par opposition, la technique de modification de la radiance ne se base pas sur des varia-

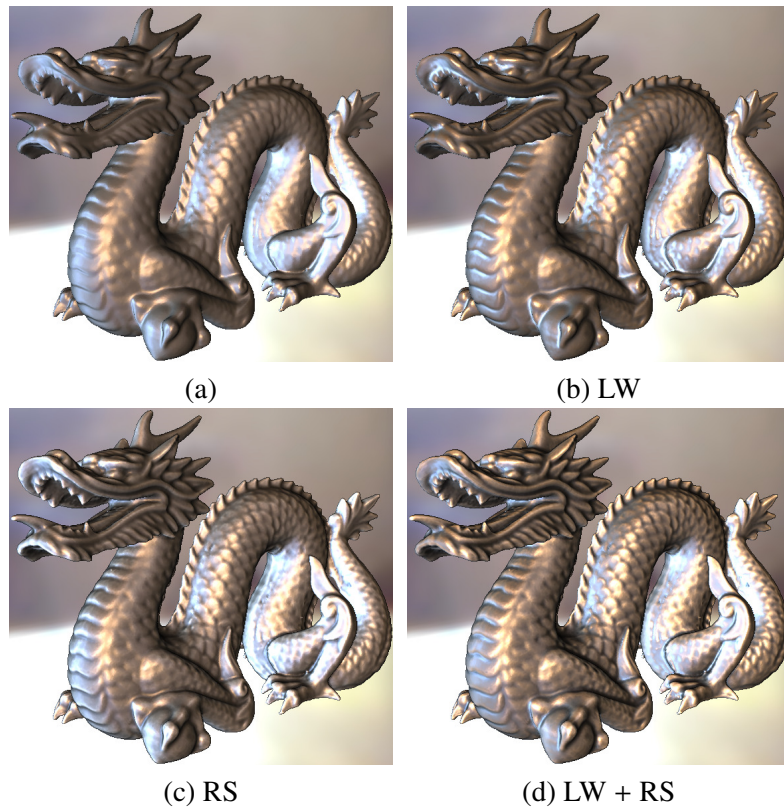


FIG. 5.26: (a) Le modèle du dragon, rendu avec un matériau brillant, sans exagération. (b) Les convexités et les concavités sont mises en valeur différemment suivant que les directions de lumière proviennent de l'avant ou de l'arrière de l'environnement. (c) La modification de la radiance fait ressortir les motifs reflétés en assombrissant/éclaircissant les concavités et convexités. (d) Combiner les deux méthodes produit une exagération plus riche : les convexités et les concavités sont mieux différenciées *et* leur contraste est plus grand.

tions dans l'environnement, mais modifie directement le rendu existant. Combiner les deux méthodes produit un résultat plus riche dans lequel tous les détails sont mis en valeur : les convexités et concavités se distinguent beaucoup plus facilement.

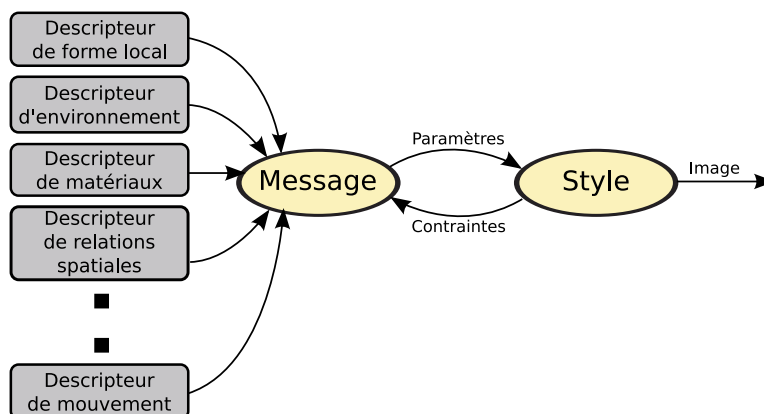
5.5 Discussion et travaux futurs

Les trois techniques présentées offrent une grande flexibilité dans la manière de mettre les détails en valeur et dans le choix des stylisations. Bien entendu, elles ne sont pas parfaites et soulèvent de nombreuses questions. Certains problèmes sont

propres à chacune des méthodes. Par exemple, LW nécessite des environnements contenant suffisamment de variations pour obtenir un résultat probant. En effet, l'exagération des détails se fait de manière indirecte avec cette méthode, et aucun effet ne serait obtenu avec des échantillons uniformes. Comme expliqué dans [FTA04], cette méthode est parfaitement adaptée aux environnements aux statistiques naturelles car elle est basée sur des résultats perceptuels. Néanmoins, il est toujours possible de la combiner avec RS, comme on vient de le voir, pour mettre en valeur les détails qui seraient masqués malgré l'exagération.

Cette seconde méthode n'est, quant à elle, pas basée sur des résultats perceptuels, mais s'avère plus flexible et plus efficace à mettre en place. Il serait donc intéressant d'étudier la manière dont nous percevons les variations et les motifs dans les images pour déterminer quelles sont les relations qui existent entre le fait de modifier l'amplitude ou la fréquence du signal. Une première approche pourrait être d'étudier ces données dans l'espace fréquentiel. Ainsi, nous pourrions voir comment le signal est modifié et pourquoi les détails sont mis en valeur.

Un autre aspect qui pourrait être étudié plus profondément est la manière de mettre en valeur les détails à la surface des objets. Jusqu'à présent, nous nous basons sur deux suppositions que nous aimerions relaxer dans le futur. (1) Les concavités et les convexités sont toujours exagérées avec des effets inverses ; et (2), l'exagération se fait avec des opérateurs de dérivation locaux seulement. Pourtant, les motifs lumineux utilisés par le SVH pour percevoir la forme des objets ne se limitent pas à une simple alternance d'intensité ou de couleurs sur les convexités et concavités [KJ03]. De nombreuses autres aspects entrent en jeu comme la direction de lumière principale [HLM06, CF07, OBA08], les caractéristiques des matériaux [Ade01, VLD07], les silhouettes [FTA04] ou encore le mouvement des objets [PS03, AVBSZ07].



F . 5.27: D'autres descripteurs pourraient être utilisés dans le but d'enrichir le message et d'obtenir une plus grande variété de styles.

Tous ces aspects pourraient être étudiés et intégrés dans le système pour améliorer le contrôle du message et augmenter le choix de styles possibles (cf. Figure 5.27). Avec l'utilisation de nouveaux descripteurs, le message pourrait être considérablement enrichi. Il serait non seulement possible de mettre en valeur les détails de forme, mais aussi d'autres caractéristiques comme les matériaux ou le mouvement dans une scène 3D. Cela laisserait aussi la possibilité d'élargir la palette de styles que nous pourrions mettre en place. Le descripteur de forme local que nous avons implémenté ne serait peut être pas le plus adapté pour orienter les coups de pinceaux dans un rendu de style peinture. Un descripteur plus global [LZ09] qui utilise une analyse en composantes principales pour trouver des directions privilégiées sur l'intégralité d'un maillage, pourrait être plus adapté.

L'utilisation d'un unique descripteur local de forme engendre aussi des problèmes avec les méthodes actuelles. Le principal est que, sans connaissance de la manière dont les détails apparaissent dans un rendu original, il est impossible de faire la distinction entre ceux qui seraient déjà clairement visibles et ceux qui seraient masqués par un éclairage contenant peu de variations. Avec un seul descripteur local, nous sommes contraints de mettre en valeur tous les détails, y compris ceux qui le seraient déjà suffisamment avec le rendu simple obtenu par l'équation de réflectance originale. La conséquence est que la perception des matériaux s'en trouve altérée lorsque l'exagération est trop forte, malgré le fait que nous ne modifions pas la géométrie des objets (en ne touchant ni aux normales, ni aux positions des sommets). Des descripteurs d'environnement et de matériaux pourraient s'avérer utiles dans ce cas, pour déterminer les surfaces qui seront déjà suffisamment bien perçues par l'utilisateur et utiliser cette information pour contrôler l'exagération.

Deuxième partie

**Représentation de la forme par le
dessin au trait**

La ligne et la forme

Le dessin au trait est un style largement utilisé dans toutes les applications, qu'elles soient artistiques ou scientifiques. Les lignes sont aujourd'hui devenues indispensables dans de nombreuses applications. Combinées ou non avec d'autres média, elles sont utilisées pour communiquer la forme, les réflexions lumineuses ou encore les expressions des visages. Elles peuvent être dessinées avec de simples courbes si l'objectif est de localiser précisément le message, ou stylisées dans certaines applications plus artistiques (jeux vidéo, films d'animation, etc.).

Le dessin au trait joue aussi un rôle fondamental dans le monde des animations car il permet aux artistes de communiquer la forme des objets et personnages avec une économie de moyens. Néanmoins, même lorsque les artistes restreignent leur dessin à un faible nombre de lignes clairement identifiées, la création d'une animation nécessite une grande patience et d'énormes compétences. La génération des lignes par ordinateur représente donc une alternative efficace : elles sont automatiquement identifiées à la surface des objets 3D, puis rendues dans des styles variés.

Après avoir vu comment les artistes communiquent la forme par des dessins au trait plus ou moins stylisés, nous étudierons deux aspects dans la suite de cette partie : le message qui correspond à l'extraction d'un ensemble de caractéristiques de surfaces représentatives des objets, puis le style qui permet de les rendre de manière cohérente spatialement et temporellement.

6.1 Les lignes créées par les artistes

Le crayon, le stylo ou la plume sont autant d'exemples d'outils utilisés par les artistes pour réaliser leurs dessins au trait. Encore une fois, les messages et styles mis en place ont des liens très forts. Dans le cas des artistes, le message peut consister en une ambiance, une expression ou encore une scène historique. Quelques exemples sont donnés dans la Figure 6.1. On peut y voir notamment le dessin d'un auroch (a), réalisé par des hommes préhistoriques, il y a environ 17000 ans. Cela

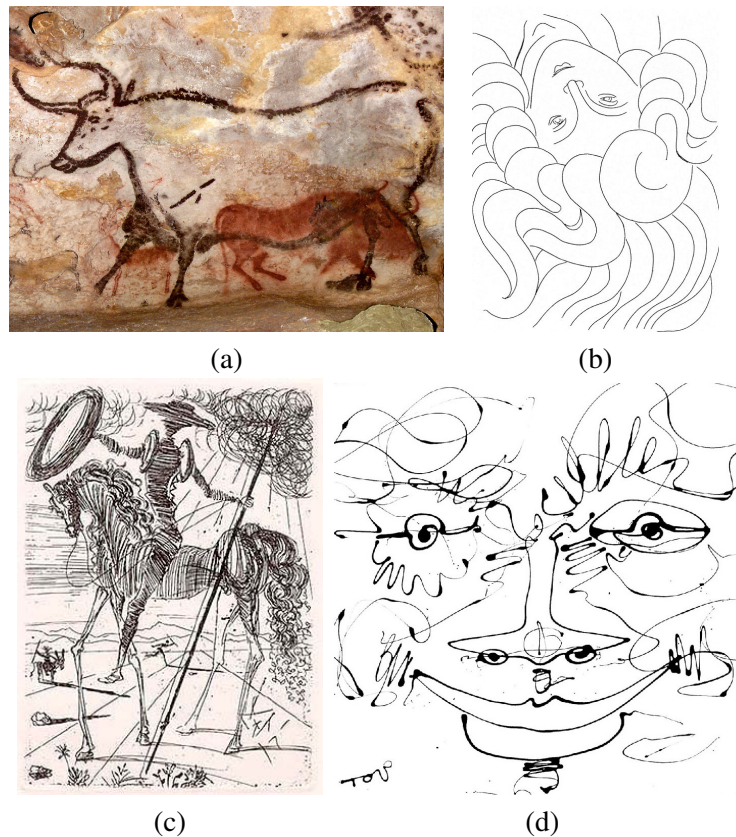


Fig. 6.1: (a) Un auroch, dessiné il y a près de 17000 ans dans la grotte de Lascaux. (b) La Chevelure - Henry Matisse, 1931-1932. (c) Don Quichotte (portrait) - Salvador Dali, 1966. (d) Le Sourire revisité - TOV.

montre à quel point les lignes représentent un moyen simple et efficace pour communiquer l'essentiel de l'information. À cette époque, le charbon de bois était l'un des pigments utilisé pour créer le noir. En ajoutant cette contrainte au fait que le support est celui du mur d'une grotte, les silhouettes de l'animal sont parfaitement représentées, ainsi que les yeux et la bouche qui sont des éléments essentiels à la compréhension de l'image.

Le dessin au trait a dès lors évolué dans des représentations plus abstraites et des styles variés. Dans l'exemple de la Figure 6.1 (b), Henry Matisse dessine la chevelure d'une femme avec des lignes très fines. Son objectif est de donner l'impression que les formes et le plan ne font qu'un. Le style choisi pour cela utilise un nombre limité de lignes très fines qui représentent la forme, de telle sorte qu'on ait l'impression qu'il n'y ait pas d'éclairage. De plus, les courbes de chacune des lignes donnent l'aspect d'une chevelure très dense. Cette technique, consistant à

former des sensations complexes avec des formes très simples dans le plan de l'image, est le principe de l'art décoratif.

Le style employé par Salvador Dali pour représenter Don Quichotte (c) est, quant à lui, très différent ; la densité et le chevauchement des lignes, combinés à des hachures, évoquent à la fois la forme et l'éclairage de la scène. Elles sont dessinées parallèlement, avec des directions privilégiées, pour différencier le personnage du cheval. Au contraire, des cercles tourbillonnants modélisent les nuages à travers desquels passent les rayons du soleil, avec des lignes droites.

Dans le dernier exemple (cf. Figure 6.1 (d)), les lignes sont utilisées dans le but de représenter l'expressivité d'un visage. Le message exprimé par l'artiste TOV est de reproduire un sourire tout en donnant une impression d'énergie et de liberté. Le peintre projette la peinture directement sur la toile posée au sol. Le hasard et la spontanéité engendrés par cette méthode procurent cette sensation de fraîcheur. Le rendu obtenu par cet équilibre instable entre concentration et hasard est assez épuré, avec des lignes courbes plus ou moins épaisses suivant les zones dessinées. Le style est ici complètement lié à ce message car la technique employée a été choisie de sorte à le renforcer.

Dans les illustrations scientifiques, le dessin au trait est l'un des styles les plus utilisés car il permet de représenter la forme des objets de manière très lisible. Dans les deux exemples de la Figure 6.2, le message des illustrateurs est de décrire au mieux la forme des objets. Le criquet dessiné par Eric Walravens a par exemple été

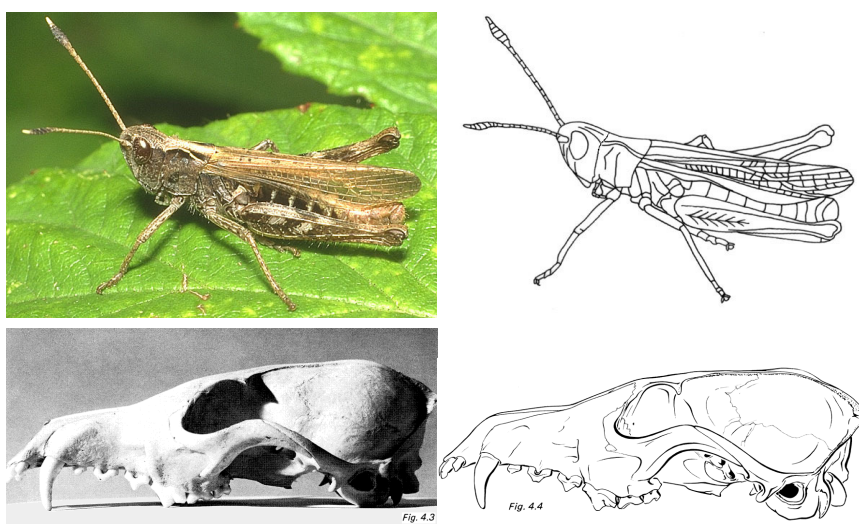


Fig. 6.2: En haut : illustration d'un criquet - Eric Walravens. En bas : illustration d'un crâne de renard - Yves Coineau.

conçu avec de nombreuses contraintes sur le style : le crayon ne doit pas être trop gras, les traits doivent être uniques, continus, jointifs, et les proportions et structures doivent être respectées pour obtenir une représentation correcte de l'insecte. L'objectif est le même dans le second exemple, où l'on peut observer le dessin d'un crâne de renard par Yves Coineau. D'autres effets intéressants apparaissent ici pour montrer les nombreux détails. On remarque notamment que les traits ne sont pas tous de la même épaisseur : des lignes fines permettent de représenter les détails fins (comme les fissures dans le crâne). D'autres, plus soutenues, comme celles que l'on retrouve au premier plan, créent une perspective aérienne et donnent du volume à l'illustration. On peut aussi noter que des détails apparaissent même dans les zones d'ombre de la photographie originale. L'objectif d'une illustration est en effet de montrer les détails sans prise en compte de l'éclairage.

L'une des contraintes à laquelle nous voulons nous confronter dans cette partie est l'animation des lignes. La principale question est de savoir comment elles se comportent lorsque le point de vue change ou lorsque les objets se déforment dans une scène. De nombreux effets peuvent en effet intervenir, comme des apparitions, des disparitions, des séparations, des regroupements ou encore des mouvements de lignes. Les artistes ont l'habitude de manipuler ces primitives, que ce soit dans



F . 6.3: À gauche : deux images de la bande dessinée “Trois Ombres” - Cyril Pedrosa. À droite : un effet de zoom sur un oiseau ; “Le Bibendum Céleste” - Nicolas Crécy.

les bandes dessinées ou dans les films d'animation. Deux exemples de zooms sont présentés dans la Figure 6.3. À gauche, les lignes du personnage sont différentes suivant qu'il se trouve proche ou loin du cadre utilisé. En zoomant sur son œil, nous distinguons nettement les narines, les oreilles, les sourcils et les rides du front qui apparaissent. D'autres effets apparaissent sur l'exemple de droite. La ligne nette située autour de l'œil de l'oiseau se divise sur la seconde vignette, où celui-ci est proche de l'observateur. Dans les films d'animation, les artistes ont le savoir-faire pour gérer la manière dont les lignes se comportent lors du mouvement des personnages. Quelques images tirées d'un film d'animation sont présentées dans la Figure 6.4. Les traits sont ici utilisés pour représenter à la fois la forme et les expressions sur le visage du personnage préhistorique. Sur les différents niveaux de détail, on peut notamment remarquer que la ligne qui forme le sourcil sur les images (a-b) se sépare lorsque le personnage est proche (c-d) pour montrer les poils. Les traits sont



FIG. 6.4: Quatre images tirées du film d’animation “Jurannessic”, créé par cinq étudiants de l’école d’animation des Gobelins en 2002 : In-Ah Rödigier, Simon Pierre Andriveau, Yann Avenati, Hervé Barbereau et Louis Clichy.

aussi beaucoup plus marqués autour des yeux sur ces dernière images. Ils permettent de reproduire l’expression du visage et certaines ombres.

La question de la cohérence temporelle des lignes est donc importante dans les animations. Les artistes ont les compétences et la patience pour les manipuler. Néanmoins, très peu de méthodes se sont penchées sur ce problème en 3D. Dans cette partie, nous présentons une nouvelle solution pour l’extraction et la stylisation de lignes sur des scènes 3D, de manière cohérente et dynamique.

6.2 Lignes définies à la surface des objets

Le premier problème auquel se sont confrontés les chercheurs a été de définir les lignes caractéristiques les plus pertinentes à la surface des objets 3D. Une attention particulière s’est portée sur l’extraction de propriétés intrinsèques sur la forme des objets. Parmi elles, les crêtes et les vallées sont souvent considérées comme des zones très importantes. Elles sont définies comme les lieux où les valeurs de courbures principales atteignent des extrema locaux dans les directions principales. Mathématiquement, ces lignes sont définies de la manière suivante :

$$\begin{aligned} e_{\max} &= 0, & \delta e_{\max} / \delta \mathbf{t}_{\max} &< 0, & \kappa_{\max} &> |\kappa_{\min}| & \text{(Crêtes)} \\ e_{\min} &= 0, & \delta e_{\min} / \delta \mathbf{t}_{\min} &> 0, & \kappa_{\min} &> -|\kappa_{\max}| & \text{(Vallées)} \end{aligned}$$

κ_{\max} , κ_{\min} sont les courbures maximales et minimales ($\kappa_{\max} \geq \kappa_{\min}$) et \mathbf{t}_{\max} , \mathbf{t}_{\min} leurs directions principales. e_{\max} et e_{\min} correspondent aux dérivées des courbures dans leurs directions respectives : $e_{\max} = \delta \kappa_{\max} / \delta \mathbf{t}_{\max}$ et $e_{\min} = \delta \kappa_{\min} / \delta \mathbf{t}_{\min}$. Ohtake *et al.* [OBS04] déterminent les crêtes et les vallées en reconstruisant localement une surface implicite à partir d’un maillage polygonal. Une méthode plus rapide, basée sur une adaptation locale du maillage par un polynôme cubique, est appliquée dans [YBS05] pour estimer le tenseur de courbure ainsi que sa dérivée. Les lignes obtenues sont souvent discontinues car elles sont calculées directement à la surface C_0 d’un maillage. Hildebrandt *et al.* [HPW05] améliorent la détection de ces lignes en ne choisissant que celles qui sont pertinentes grâce à un filtre Laplacien appliqué sur la surface.

Si les crêtes et les vallées sont définies comme étant les lieux où la dérivée s'annule et change de signe, les lignes de démarcation [KST08] sont les points pour lesquels la courbure change de signe dans la direction du gradient de la courbure. Intuitivement, il s'agit des lignes entre les crêtes et les vallées, qui séparent les zones convexes et concaves à la surface des objets. Les arêtes de relief définies par Kolomenkin *et al.* [KST09] permettent de réduire le bruit pouvant être généré par les lignes de démarcation lorsque la courbure varie beaucoup. Elles sont calculées comme les lieux où la courbure s'annule et change de signe dans la direction perpendiculaire à l'arête. De nombreuses autres lignes peuvent ainsi être définies à la surface des objets. Parmi elles, on peut citer les lignes paraboliques qui sont les points pour lesquels la courbure gaussienne est nulle, ou encore les points d'inflexion (qui se rapprochent des lignes de démarcation) et qui correspondent aux points pour lesquels la courbure moyenne est nulle [Koe90]. La Figure 6.5 montre quelques exemples de lignes obtenues avec ces méthodes sur un objet archéologique.

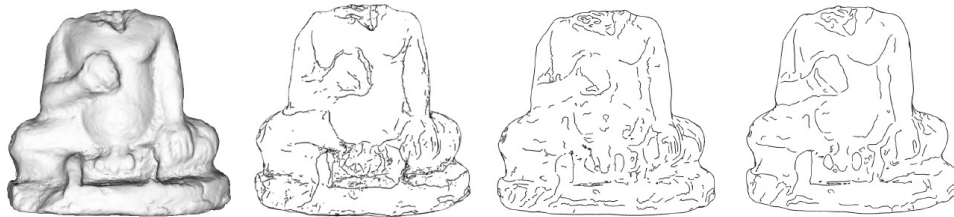


Figure 6.5: De gauche à droite : objet original ; crêtes et vallées [OBS04] ; lignes de démarcation [KST08] ; arêtes de relief [KST09]. Image tirée de [KST09].

Les caractéristiques intrinsèques que nous venons de voir sont utiles pour communiquer des propriétés purement géométriques de la surface des objets et peuvent être précalculées pour ensuite être employées dans les applications. Elles ne sont néanmoins pas toujours adaptées dans l'optique de la stylisation, car elles ne prennent en compte ni le point de vue, ni l'éclairage. Les lignes dépendantes du point de vue les plus connues sont assurément les occultations. Elles correspondent aux points de la surface pour lesquels la normale et la direction de la caméra sont perpendiculaires (i.e. $\mathbf{n} \cdot \mathbf{e} = 0$) et représentent les silhouettes des objets ainsi que les frontières entre différentes parties connexes depuis un point de vue donné. La méthode classique (mais laborieuse) consiste donc à parcourir le maillage pour chaque nouvelle image, puis de garder les arêtes qui séparent les polygones orientés vers l'observateur des polygones orientés dans la direction opposée. Markosian *et al.* [MKG⁺97] ne testent que les arêtes déjà trouvées précédemment pour éviter de tout parcourir à chaque changement de point de vue. Des algorithmes plus

simples et plus rapides ont été proposés par Raskar et Cohen pour détecter les silhouettes en deux passes [RC99] (en calculant l'intersection des faces visibles et cachées en espace image), puis en une seule passe [Ras01] avec l'utilisation des fonctionnalités des cartes graphiques. D'autres encore permettent de détecter les silhouettes dans un moteur de lancer de rayons [CP09].

Il existe un grand nombre de définitions de lignes dépendantes du point de vue liées à la courbure. Les contours et reflets suggestifs, introduits par DeCarlo *et al.* [DFRS03, DR07], représentent les points pour lesquels la courbure radiale est nulle et permettent d'étendre les silhouettes. La courbure radiale est la courbure dans la direction de vue projetée sur le plan tangent. Intuitivement, ces points correspondent à des silhouettes et à leur prolongement pour des directions de vue proches de celle courante. La technique de base peut être modifiée pour améliorer la cohérence temporelle et éviter l'apparition abrupte de lignes [DFR04]. De leur côté, Judd *et al.* [JDA07] introduisent la notion de crête apparente en utilisant un calcul de courbure dépendant du point de vue (cf. Chapitre 4). Plus stable que les contours suggestifs, leur méthode consiste à appliquer la définition originale des crêtes en projetant au préalable le tenseur de courbure en espace image pour travailler sur des données prenant le point de vue en compte. Des définitions de lignes alternatives prennent la direction de la lumière en compte pour définir des lignes spécifiques. Les lignes photiques extrémales [XHT⁺07] sont par exemple définies comme l'ensemble des points d'une surface 3D pour lesquels l'intensité de l'éclairage atteint un maximum local dans la direction de son gradient. Les lignes laplaciennes, introduites par Zhang *et al.* [ZHXC09], moins complexes à mettre en œuvre, sont les *zero-crossings*¹ du Laplacien de l'intensité de la surface. Quelques exemples de ces lignes sont donnés dans la Figure 6.6.



F . 6.6: De gauche à droite : silhouettes ; contours suggestifs [DFRS03] ; crêtes apparentes [JDA07] ; lignes laplaciennes [ZHXC09]. Image tirée de [ZHXC09].

Même en introduisant des fonctionnalités dépendantes du point de vue ou de la lumière, toutes les méthodes que nous venons de voir nécessitent des précalculs à la surface des objets 3D (excepté les occultations). Les lignes sont ensuite extraites en temps-réel en effectuant des calculs simples à partir de ces données. Le principal inconvénient de l'ensemble de ces techniques est qu'elles ignorent complètement

¹Les lieux où la fonction considérée s'annule et change de signe.

le profil des détails qui sont détectés, après qu'ils aient été projetés sur l'écran (i.e. la forme de la surface située sous les lignes détectées). La conséquence directe est que les lignes ne sont détectées qu'à une échelle unique ; aucun nouveau détail n'apparaît lorsque la caméra se rapproche de la surface des objets, et à l'opposé, les lignes s'agglomèrent et s'encombrent lorsque les objets sont éloignés du point de vue, sans se mélanger correctement. De plus, les précalculs nécessaires rendent ces méthodes mal adaptées aux objets déformables. Certaines techniques permettent néanmoins de précalculer les données requises comme les intensités et directions de courbures sur des échelles multiples [NJLM06, CJG09]. D'autres précalculent ces informations pour des intervalles de déformation [KNS⁺09]. Mais les solutions proposées ne permettent de résoudre le problème que de manière partielle : elles ne permettent pas de gérer des scènes dynamiques efficacement car elles nécessitent des interventions de l'utilisateur et des précalculs coûteux en temps et en mémoire.

6.3 Stylisation de lignes paramétrées

L'un des principaux avantages que procurent les lignes extraites en espace objet est leur facilité de stylisation. En travaillant directement sur le maillage, les lignes sont généralement obtenues sous la forme d'une série de points 3D qui sont ensuite reliés entre eux. Avec cette définition explicite, il est assez simple d'obtenir des lignes très précises, sans aliasing [CF08], ou d'utiliser la transparence pour afficher les lignes visibles et celles qui ne le sont pas lors du rendu [CF09]. Des fonctions peuvent facilement être appliquées sur les vecteurs de points pour obtenir des styles variés. Dans [GVH07], la largeur des traits est choisie en fonction de paramètres comme la courbure ou la profondeur. La méthode de stylisation la plus répandue consiste à plaquer des textures le long des lignes projetées sur l'écran. Grâce à la définition explicite des lignes, il est en effet possible de générer facilement des paramétrisations le long desquelles des fonctions sont appliquées. Grabli *et al.* [GTDS04, GTDS10] laissent par exemple la possibilité à l'utilisateur de programmer leurs propres styles le long des lignes (Figure 6.7 (a)). Les caractéristiques de stylisation peuvent être associées à des propriétés géométriques comme la profondeur, la courbure, la couleur, ou l'orientation de la surface.

Mais si l'espace des styles créés peut être varié, ces méthodes ne peuvent néanmoins être appliquées que sur des images fixes. La raison de cette limitation est que la paramétrisation est réévaluée à chaque image et n'est pas forcément cohérente dans le temps. Du bruit, des scintillements peuvent survenir lorsque les objets bougent et que des changements topologiques apparaissent et modifient la paramétrisation. Des méthodes spécifiques ont alors été implémentées dans le but d'animer ces lignes (et principalement les silhouettes) en limitant les problèmes de cohérence temporelle. Des textures procédurales sont utilisées dans [KMN⁺99, MMK⁺00] pour que le style soit préservé lorsque les objets s'éloignent de la caméra. Des chaînes de triangles créées le long des lignes et sur lesquels des tex-

tures sont plaquées sont aussi mises en place dans la même optique dans [NM00] (Figure 6.7 (b)). La transparence est utilisée lors de changements topologiques de sorte que des transitions douces soient effectuées entre les styles. Le système WYSIWYG NPR² de Kalnins *et al.* [KMM⁺02] permet de synthétiser les textures le long des silhouettes et arêtes vives dans le but de rendre la création de nouveaux styles plus facile (Figure 6.7 (c)). Néanmoins, ces méthodes ne résolvent pas tous les problèmes de cohérence temporelle et de nombreux artefacts persistent. Kalnins *et al.* [KDMF03] proposent une solution pour contrecarrer ces incohérences en suivant et en optimisant la paramétrisation sur chaque nouvelle image pour limiter les changements abrupts qui peuvent apparaître notamment lors de compressions ou de dilatations. Bénard *et al.* [BCGF10] traitent le même problème en mettant à jour, non pas la paramétrisation, mais les textures qui possèdent des propriétés de similarité que l'on retrouve dans de nombreux styles (Figure 6.7 (d)). Ces dernières méthodes de stylisation parviennent à améliorer la cohérence

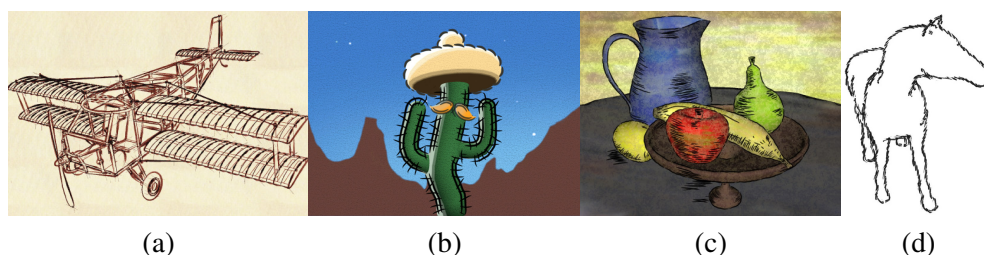


FIG. 6.7: Quatre résultats où des textures sont utilisées pour styliser les lignes en disposant d'une paramétrisation. (a) Style programmable de Grabli *et al.* [GTDS10]. (b) Silhouettes artistiques [NM00]. (c) WYSIWYG NPR [KMM⁺02]. (d) Utilisation de textures cohérentes de Bénard *et al.* [BCGF10].

temporelle sur des lignes stylisées paramétrées qui s'allongent ou rétrécissent au cours du temps. Néanmoins, des problèmes d'incohérence subsistent généralement lorsque les lignes se recollent ou se séparent. Cela est dû aux changements soudains de paramétrisations qui peuvent survenir. Si ces problèmes ne sont pas très gênants pour la stylisation de silhouettes car les lignes se séparent ou se recollent principalement à leurs points terminaux, ils peuvent être beaucoup plus problématiques pour les autres types de lignes. En considérant par exemple deux lignes de crêtes parallèles qui s'assemblent sur la surface d'un objet lorsque celui-ci s'éloigne de la caméra, il ne semble pas y avoir d'approche naturelle pour paramétrer les nouvelles configurations sans créer de discontinuité temporelle.

²Pour *What You See Is What You Get Non Photorealistic Rendering*.

6.4 L'approche image

Une alternative à la stylisation des lignes est de les étudier en espace image. Il est bien entendu possible d'utiliser les mêmes techniques de paramétrisation pour styliser des images fixes [SKLL07]. Mais dans ce cas, des techniques spécifiques doivent aussi être mises en place pour simplifier et garder une certaine densité de lignes lorsque les images reculent ou rétrécissent [BTS05, SC08]. L'approche par filtrage (Sobel, Prewitt, Laplace, Canny [Can86], etc.) est, quant à elle, très différente dans le sens où les lignes sont obtenues de manière implicite ; on ne possède pas de vecteurs de points, mais seulement une propriété locale, qui permet de déterminer si chacun des pixels se trouve sur une ligne caractéristique ou non.

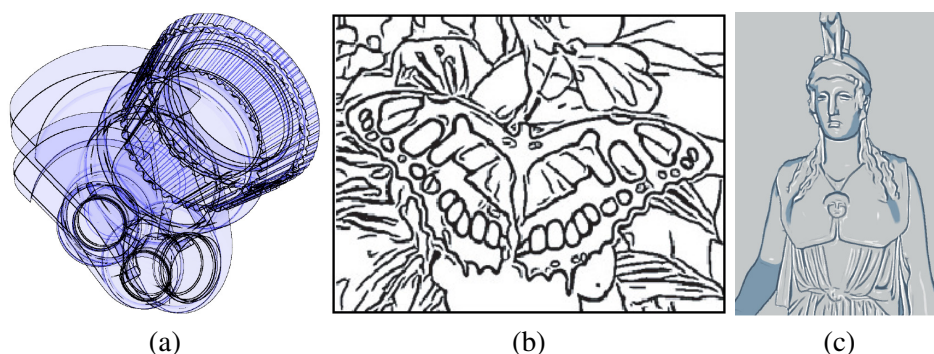


FIG. 6.8: Exemples de détections et de stylisations de lignes en espace image. (a) Utilisation de la transparence [ND04]. (b) Détection de lignes cohérentes [KLC07]. (c) Détection des crêtes sur la luminosité de Lee *et al.* [LMLH07].

Les premiers à appliquer cette technique sur des données géométriques 3D projetées sur l'écran sont Saito et Takahashi [ST90]. Ils introduisent les G-Buffers qui sont des textures permettant de stocker des propriétés géométriques projetées en espace image, puis y appliquent des filtres spécifiques pour extraire les lignes. Ils montrent qu'il est ainsi possible de détecter les occultations et les arêtes vives en appliquant des filtres simples sur les textures de profondeurs et de normales. Cette méthode a ensuite été adaptée sur le GPU par Niehaus et Döllner [ND04] (Figure 6.8 (a)). Ils ajoutent la gestion de la transparence avec du *depth peeling* (i.e. des plans de coupe sont utilisés pour générer des textures avec plusieurs niveaux de visibilité) et un effet de distorsion des lignes grâce à du bruit créé en espace image. Cette même technique est aussi utilisée dans [IB06] pour sauvegarder les propriétés géométriques des lignes préalablement calculées en espace objet. La particularité de ces méthodes de filtrage est que le message (i.e. la définition de la ligne) et le style sont tous deux définis par le filtre lui-même. L'intensité et la couleur des lignes sont obtenues avec l'application du filtre, et l'épaisseur est créée de manière indirecte. Par exemple, des dérivées de gaussiennes sont appliquées le long du flot des

tangentes dans [KLC07, KLC09] pour obtenir des lignes très propres qui ont toutes à peu près la même épaisseur (Figure 6.8 (b)). L'inconvénient de ce lien très fort entre message et style est que le choix du filtre limite très fortement les détails qui sont extraits ainsi que les propriétés des lignes créées, comme leur épaisseur. Pour le moment, la seule approche en espace image permettant de contrôler l'épaisseur des lignes en fonction du profil de la surface est celle de Lee *et al.* [LMLH07]. Leur méthode permet d'extraire les arêtes vives et les crêtes sur le rendu d'objets en faisant une approximation locale de la luminance avec une fonction quadratique (Figure 6.8 (c)). Néanmoins, cette technique est limitée à l'extraction des détails de luminance seulement et ne peut produire que des styles très simples.

6.5 Une nouvelle vision du dessin au trait

L'ensemble des méthodes présentées, qu'elles soient en espace objet ou image, possède des avantages et des inconvénients. On peut se référer à [IFH⁺03, RDF08] pour des états de l'art plus poussés. L'extraction explicite des lignes à la surface des objets permet de les paramétrer et de les styliser facilement. Cette méthode est néanmoins difficilement adaptable aux scènes animées : il n'est souvent pas possible de précalculer les données nécessaires pour toutes les poses existantes, la paramétrisation (et donc le style) peut changer de manière abrupte, et le profil des détails n'est pas pris en compte lors de la stylisation (ce qui provoque des problèmes pour les niveaux de détail). En espace image, les méthodes par filtrage ont l'avantage d'extraire les lignes dynamiquement en fonction de la projection des objets. Par contre, la variété de styles est très limitée car le filtre définit à la fois le message et le style. La seule propriété pouvant être vraiment contrôlée est l'épaisseur des lignes [LMLH07].

Dans la suite de cette partie, nous proposons une nouvelle approche permettant de détecter et de styliser des lignes en espace image. Notre objectif est de répondre aux deux challenges qui représentent les limitations des approches précédentes :

1. extraire un ensemble de lignes de manière dynamique et cohérente,
2. obtenir une variété de styles qui se comportent bien lorsque des changements topologiques surviennent.

Nous présentons ainsi une alternative aux approches existantes que nous appelons *Brosses Implicites* (BI). L'idée principale est de parvenir à résoudre ces problèmes en utilisant une approche implicite entièrement en espace image. A chaque nouvelle image, notre système identifie les lignes ainsi que leurs profils (le message qui sera décrit dans le Chapitre 7). Il permet ensuite de styliser les lignes par un procédé de convolution pour obtenir un rendu final cohérent spatialement et temporellement (Chapitre 8).

De cette manière, les lignes stylisées apparaissent et disparaissent de manière cohérente pour communiquer des informations caractéristiques sur la forme de la surface des objets. Comme il n'y a pas besoin de paramétrisation, toutes les distorsions qui peuvent se produire sont gérées naturellement, image par image, sans avoir à suivre explicitement les détails de surface. Cette approche fonctionne non seulement en temps-réel sur des scènes dynamiques (avec des objets déformables qui peuvent changer de topologie), mais elle est aussi indépendante de la complexité de la scène 3D. De plus, elle fonctionne avec des définitions variées de lignes comme les occultations, les crêtes, les vallées, les points d'inflexion et pourrait s'incorporer facilement dans un logiciel de composition vidéo. En travaillant seulement en espace image, il est aussi possible d'appliquer la technique sur des vidéos. Les résultats sont comparables à ceux que l'on pourrait obtenir avec des logiciels tels que Photoshop ou Gimp, dans lesquels il est possible de dessiner directement avec des brosses stylisées, par opposition aux logiciels basés sur des définitions explicites des vecteurs comme Illustrator ou Inkscape.

Généralisation de la définition d'une ligne

Comme il a été vu dans le chapitre précédent, la plupart des approches existantes se limitent soit à l'extraction du message, soit à la stylisation (excepté les filtres appliqués en espace image). D'après des études récentes, il a été montré que toutes les définitions de lignes ne permettent pas d'apprécier la forme de la même manière [CSD⁺09]. De plus, lorsqu'il est demandé à des artistes de dessiner des scènes avec de simples traits, ceux-ci ne se limitent pas à une seule définition

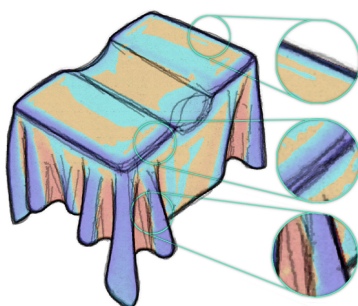


Fig. 7.1: Cette illustration, dans laquelle nous avons combiné notre descripteur local de forme (convexités en bleu, concavités en orange) avec des traits dessinés par des artistes (en noir), montre que les lignes peuvent communiquer des informations différentes sur la forme. Les dessins des sujets ont été accumulés pour montrer la variété des résultats, et montrent qu'ils dessinent des extrema et des inflexions. Les corrélations entre les personnes diffèrent en fonction du profil des détails : toutes les lignes concordent fortement sur les occultations (en haut à droite), sont un peu étalées le long des détails fins (au milieu) et vaguement représentées ou même omises sur des surfaces lisses (en bas à droite).

[CGL⁺08]. Il n'y a donc pas de consensus sur le type des lignes à extraire à la surface des objets. En particulier, les lignes dessinées ne représentent pas uniquement les lieux où la courbure est la plus forte (les crêtes et les vallées), mais peuvent aussi souligner les transitions entre convexités et concavités (les points d'inflexion). Dans l'illustration de la Figure 7.1, nous avons combiné le descripteur de forme local (défini dans le Chapitre 4) avec des dessins au trait (en noir) réalisés par différentes personnes et issus de l'étude de Cole *et al.* [CGL⁺08]. Il semble que les lignes dessinées par les sujets soient de moins en moins corrélées lorsque la surface est de moins en moins courbée. Lorsque celle-ci est trop lisse et se rapproche d'un plan, de moins en moins de lignes apparaissent, jusqu'à ce qu'elles disparaissent totalement. La seule exception se trouve cependant le long des occultations qui représentent des discontinuités de visibilité très fortes dans le rendu. Ces observations suggèrent qu'en moyenne, les lignes représentent des détails dont le profil est suffisamment courbé. De plus, ce profil évolue durant l'animation d'une scène, lorsque les objets subissent des déformations, des rotations, ou que la caméra se rapproche ou s'éloigne des surfaces. C'est la raison pour laquelle les lignes doivent être extraites de manière dynamique, à chaque nouvelle image. L'extraction du message et sa stylisation que nous présentons dans ce chapitre et le suivant sont en cours de soumission.

7.1 Identification des lignes

La technique d'extraction des détails que nous avons mise en place n'a pas pour objectif de définir de nouveaux types de lignes, mais plutôt de généraliser la technique implicite de Lee *et al.* [LMLH07] en fonctionnant avec des définitions de lignes arbitraires et en fournissant un intervalle de stylisation beaucoup plus grand. Le descripteur local de forme que nous avons mis en place au Chapitre 4 n'est pas suffisant pour détecter tous ces types de lignes. Par exemple, choisir directement la courbure moyenne pour créer le rendu final limiterait fortement le message (il ne serait possible de ne représenter que la courbure) mais aussi le style : nous n'aurions pas suffisamment d'informations pour contrôler l'épaisseur ou d'autres propriétés nécessaires. Il nous faut donc améliorer le descripteur de sorte à avoir des informations pertinentes sur chacun des pixels.

7.1.1 Squelette des détails

En regardant en détail les différentes définitions de lignes existantes, nous avons constaté qu'elles consistent toutes à détecter des maxima locaux d'invariants géométriques dans une direction tangente donnée. Par exemple, les crêtes sont définies comme étant les maxima locaux de la courbure maximale dans la direction de courbure maximale. Si cette définition est à l'origine implémentée en espace objet, des caractéristiques similaires sont obtenues en espace image avec tous les avantages que nous avons démontrés : dépendance du point de vue, niveaux de détail automatiques, séparation et agglomération naturelle des lignes, etc. Dans

la suite de ce chapitre, on appelle *squelette* les points sur lesquels surviennent ces maxima locaux. Il est formellement défini par :

$$\mathcal{S} = \left\{ \mathbf{s} \in \mathbb{R}^2 \mid \frac{\delta h(\mathbf{s})}{\delta \theta(\mathbf{s})} = 0, \frac{\delta^2 h(\mathbf{s})}{\delta \theta(\mathbf{s})^2} < 0 \right\}. \quad (7.1)$$

où $\mathcal{S} \subset \mathbb{R}^2$, $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ est un champ de hauteurs C^2 et $\theta : \mathbb{R}^2 \rightarrow \mathbb{S}^2$ est un champ de directions C^1 , définis dans le plan image. h et θ peuvent être facilement instanciés pour produire la plupart des lignes existantes de manière générique.

Les détails d'ordre 1 sont obtenus en calculant les gradients en espace image, comme nous l'avons vu pour le descripteur dans le Chapitre 4. Les contours peuvent être approchés en prenant les maxima du gradient de la profondeur \mathbf{g}_d dans sa direction correspondante. Les arêtes de surface sont obtenues en prenant les maxima du gradient de la surface $\mathbf{g}_n = (-n_x/n_z, -n_y/n_z)$ dans sa direction correspondante, où $\mathbf{n} = (n_x, n_y, n_z)$ est, comme nous l'avons déjà vu, la normale de la surface exprimée en espace écran. Cette définition est similaire à celle des contours suggestifs calculés en espace image de DeCarlo *et al.* [DFRS03] : ils sont définis comme les minima de $(\mathbf{n} \cdot \mathbf{e})$. Les arêtes de luminance sont calculées en prenant les maxima du gradient de la luminance \mathbf{g}_l dans sa direction correspondante. Cette définition est similaire à celle de Lee *et al.* [LMLH07] et peut être vue comme une formulation en espace image des lignes photiques extrémales [XHT⁺07].

Les crêtes et les vallées sont des caractéristiques d'ordre 2, et nécessitent donc un tenseur de courbure \mathbf{II} avec lequel il est possible d'extraire les intensités de courbures principales κ_{\max} et κ_{\min} ainsi que leurs directions correspondantes \mathbf{t}_{\max} et \mathbf{t}_{\min} . L'extraction de ces données se fait très facilement à partir du descripteur local défini en espace image et présenté dans le chapitre 4. Les crêtes sont obtenues avec les maxima du champ de hauteurs κ_{\max} dans le champ de directions \mathbf{t}_{\max} . Même chose pour les vallées avec les maxima de $-\kappa_{\min}$ dans la direction \mathbf{t}_{\min} . Cette définition est qualitativement équivalente à celle donnée dans [JDA07], mais possède de nombreux avantages : les directions de courbures de notre tenseur sont orthogonales, ce qui fait que les vallées sont aussi bien définies. De plus, utiliser un voisinage en espace image pour les calculs évite l'agglomération des détails lorsque la caméra s'approche ou s'éloigne de la scène.

Enfin, les points d'inflexion représentent des détails d'ordre 3 et nécessitent de calculer une information de variation de courbure. Pour les calculer, nous prenons les maxima du gradient de la courbure moyenne \mathbf{g}_H dans sa direction correspondante. Ces lignes sont similaires aux courbes de démarcation [KST08] avec une définition en espace image. L'ensemble de ces définitions de lignes est décrit dans la Table 7.1.

Nom	h	θ
Occultations	$ \mathbf{g}_d $	$\mathbf{g}_d/ \mathbf{g}_d $
Arêtes de surface	$ \mathbf{g}_n $	$\mathbf{g}_n/ \mathbf{g}_n $
Arêtes de luminance	$ \mathbf{g}_l $	$\mathbf{g}_l/ \mathbf{g}_l $
Crêtes Apparentes	κ_{\max}	\mathbf{t}_{\max}
Vallées Apparentes	$-\kappa_{\min}$	\mathbf{t}_{\min}
Inflexions	$ \mathbf{g}_H $	$\mathbf{g}_H/ \mathbf{g}_H $

Tab. 7.1: Liste des définitions de lignes les plus répandues. $\mathbf{g}_{d,n,l}$ sont respectivement les gradients de la profondeur, de la normale, et de la luminance. $\kappa_{\min,\max}$ sont les courbures minimales et maximales, et $\mathbf{t}_{\min,\max}$ sont leurs directions respectives. Enfin, \mathbf{g}_H correspond, dans notre cas, à la variation de la courbure moyenne.

7.1.2 Profil des détails

L'avantage que procure l'utilisation de l'équation 7.1 est que nous pouvons maintenant raisonner sur des détails abstraits, sans nous focaliser sur une définition particulière. Il nous suffit en effet d'un champ de hauteurs et d'un champ de directions pour détecter les lignes ainsi que leurs caractéristiques (que nous appelons *profils*). Nous illustrons notre technique sur une surface ondulée. Dans la Figure 7.2, les courbes rouges représentent le squelette de la fonction h . Cet exemple montre bien à quel point il est important de détecter plus d'informations que le squelette seul, car nous souhaitons pouvoir communiquer les différences d'épaisseurs et d'intensités entre les oscillations. Une autre observation est que toutes les

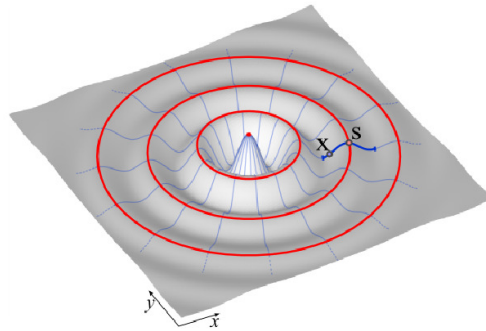


Fig. 7.2: Exemple simple d'une surface ondulée. On montre le champ de hauteurs $h(\mathbf{x}) = \cos(|\mathbf{x}|)/(1+0.2|\mathbf{x}|)$ (rendu avec une simple fonction Lambertienne ici), un sous ensemble (en bleu clair) des trajectoires du champ de directions $\theta(\mathbf{x}) = \mathbf{x}/|\mathbf{x}|$, le squelette correspondant \mathcal{S} des détails qui dans ce cas sont les crêtes (en rouge) et un profil caractéristique (en bleu foncé) passant par les points $\mathbf{x} \notin \mathcal{S}$ et $\mathbf{s} \in \mathcal{S}$.

informations utiles pour faire ces distinctions sont contenues dans les champs de directions et de hauteurs. En effet, Do Carmo [dC76] montre que pour chaque point non singulier \mathbf{x} d'un champ de directions θ , il existe une courbe unique $c_{\mathbf{x}}(t)$ qui passe par \mathbf{x} et dont la tangente est partout égale à θ . Une telle courbe est appelée une *trajectoire* (ou courbe intégrale). Un sous ensemble des ces trajectoires est dessiné en bleu clair sur la Figure 7.2. Cependant, il est possible qu'une trajectoire traverse plusieurs fois le squelette d'un détail \mathcal{S} . Pour identifier l'unique profil correspondant au point $\mathbf{s} \in \mathcal{S}$, nous restreignons sa trajectoire aux minima (ou singularités) du champ de hauteurs sur chacun des cotés de \mathbf{s} . Cet intervalle est illustré dans la Figure 7.2 par la courbe bleu foncée. Le profil $h_{\mathbf{s}} :]t_-, t_+[\rightarrow \mathbb{R}$ d'un point $\mathbf{s} \in \mathcal{S}$ correspond à la fonction de hauteur le long de cette trajectoire tronquée :

$$p_{\mathbf{s}}(t) = h \circ c_{\mathbf{s}}(t), \quad t \in]t_-, t_+[, \quad (7.2)$$

où t_+ et t_- sont respectivement les positions paramétriques positives et négatives des deux plus proches minima ou singularités du champ de hauteurs.

Une propriété intéressante de l'équation 7.2 est qu'elle s'applique aussi sur tous les points non singuliers et non-minimaux $\mathbf{x} \notin \mathcal{S}$. Toutefois, comme toutes les trajectoires sont uniques, pour chaque point \mathbf{x} se trouvant sur le profil centré sur un point \mathbf{s} (comme dans la Figure 7.2), $p_{\mathbf{s}}(t)$ et $p_{\mathbf{x}}(t)$ sont équivalents à une translation paramétrique. En d'autres termes, le squelette et le profil d'un détail sur le champ de hauteurs peuvent être obtenus de manière implicite autour de \mathbf{x} en analysant un voisinage le long de sa trajectoire.

7.2 Implémentation

Nous utilisons cette propriété pour extraire les détails de manière parallèle sur les cartes graphiques récentes. L'extraction des détails, définie de manière continue, s'adapte facilement au domaine discret de l'image. Cela se fait en trois étapes : (1) nous calculons tout d'abord $h(\mathbf{x})$ et $\theta(\mathbf{x})$ par pixel avec des opérateurs d'analyse en espace image ; (2) nous construisons un voisinage 1D autour de chacun des pixels en suivant leurs trajectoires ; (3) nous évaluons le squelette et le profil de ces fonctions par pixel en approchant localement la surface à l'aide d'une fonction analytique (i.e. avec une procédure de *fitting*).

7.2.1 Extraction des données

Pour calculer les propriétés d'ordre 1, nous obtenons les différents gradients en appliquant un filtre de Sobel sur la texture appropriée : la carte de profondeurs pour les occultations et la carte de luminances (en moyennant les couleurs) pour les arêtes de luminance. Le gradient de la surface $\mathbf{g}_{\mathbf{n}}$ requis pour les arêtes de surface est directement obtenu par les normales projetées en espace image, comme expliqué dans la section précédente. Pour les définitions impliquant des calculs

d'ordre 2, le descripteur local de forme est parfaitement adapté pour obtenir le gradient de \mathbf{g}_n . Enfin, les caractéristiques d'ordre 3 sont calculées en appliquant un filtre de Sobel sur la courbure moyenne H , elle-même extraite du tenseur de second ordre \mathbf{II} .

Mis à part pour les silhouettes qui représentent des discontinuités de profondeur très nettes, les filtres que nous appliquons en espace image doivent pouvoir prendre en compte des voisinages de tailles variées pour sélectionner des détails plus ou moins larges. Pour cela, nous appliquons une diffusion anisotrope sur le gradient de la surface (ou de la luminance), comme pour le calcul de notre descripteur dans la Section 4.3, en prenant soin de ne pas diffuser sur des régions non connexes (i.e. en prenant les silhouettes comme isolateurs). Ce procédé permet de lisser non seulement le champ de hauteurs h , mais aussi le champ de directions θ , de sorte qu'on ne sélectionne que des détails larges en gardant la cohérence des lignes.

Une fois que nous avons identifié h et θ sur chacun des pixels pour un choix particulier de détail (cf. Table 7.1), il nous faut localiser les singularités du champ de vecteurs avant d'estimer le profil. En effet, il est possible que ce champ ne soit pas défini sur certains points, dans le cas où le gradient est nul ou si les directions ne sont pas cohérentes (par exemple sur les points ombiliques lorsque l'on s'intéresse à la direction de courbure maximum). Les singularités de θ sont approchées en examinant la variation angulaire moyenne de la direction de chacun des pixels \mathbf{x} avec celles de leurs huit voisins :

$$\gamma_{\theta}(\mathbf{x}) = 1 - \sum_{i=1}^8 |\theta(\mathbf{x}) \cdot \theta_i(\mathbf{x})| / 8$$

$\theta_i(\mathbf{x})$ étant l'orientation du pixel i . Les occultations doivent aussi être considérées comme des singularités car elles délimitent des voisinages de surface non connexes. Elles sont approchées par $\gamma_d(\mathbf{x}) = \|\mathbf{g}_d(\mathbf{x})\|$. Les singularités sont alors identifiées comme l'union des singularités dues aux contours et aux silhouettes :

$$\gamma(\mathbf{x}) = \max(\gamma_{\theta}(\mathbf{x}), \gamma_d(\mathbf{x}))$$

Les données que nous obtenons par pixel sont illustrées dans la Figure 7.3 : h est représenté par une image en niveaux de gris et la technique de *Line Integral Convolution* (LIC) [CL93] permet d'afficher les directions θ avec les singularités surimposées en rouge.

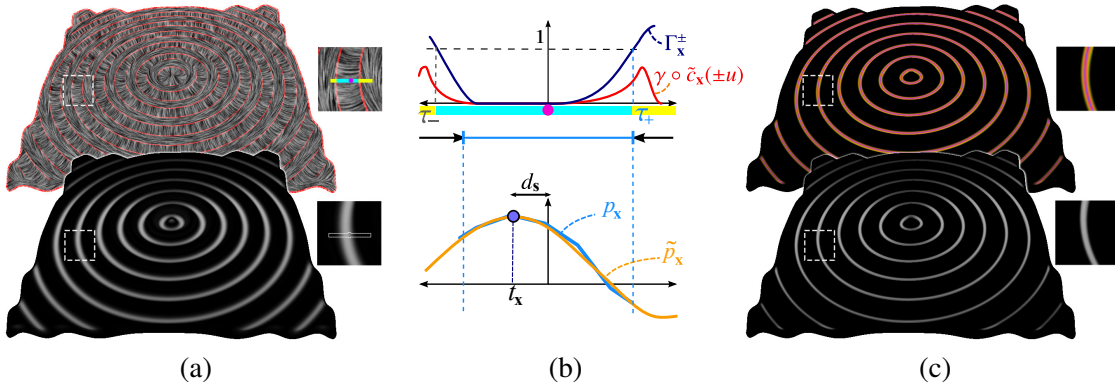
7.2.2 Échantillonnage du profil

La seconde étape de notre analyse consiste à prendre avantage de l'observation qui a été faite à la fin de la Section 7.1 : comme chaque pixel \mathbf{x} non singulier appartient à une trajectoire unique $c_{\mathbf{x}}(t)$, nous pouvons parcourir $c_{\mathbf{x}}(t)$ pour découvrir le profil de la surface, ainsi que le point qui correspond à son squelette. Ce procédé est néanmoins sujet à des problèmes numériques car il se peut que les champs de

vecteurs et de hauteurs h et θ contiennent du bruit. L'évaluation des dérivées directionnelles premières et secondes sur des surfaces telles que des maillages rend en effet le calcul de \mathcal{S} sensible au bruit (dû à l'aliasing ou à la tessellation). Pour résoudre ce problème, nous considérons une expansion de Taylor de premier ordre pour parcourir la trajectoire $c_x(t) : \tilde{c}_x(t) = \mathbf{x} + t \theta(\mathbf{x})$. Cette approximation, qui prend en compte un voisinage linéaire, est cohérente et valide pour les points se trouvant à proximité du squelette \mathcal{S} car nous avons observé que les trajectoires se rapprochent de simples lignes droites. Dans notre système, nous mesurons $p_x(t)$ le long de $\tilde{c}_x(t)$, avec $2k + 1$ échantillons (que l'on nomme $t_i, i = -k..k$) distribués uniformément sur chacun des cotés de \mathbf{x} (nous utilisons $k = 4$ dans notre implémentation).

Il nous faut aussi faire attention à ne pas traverser les singularités lorsque nous échantillons le voisinage. Pour éviter cela, nous prenons une approche similaire à la diffusion anisotrope : nous réduisons $\tilde{c}_x(t)$ lorsque l'on se rapproche d'une singularité. Pour cela, nous accumulons tout d'abord les valeurs de γ sur chacun des cotés de \mathbf{x} :

$$\Gamma_{\mathbf{x}}^{\pm}(t_i) = \sum_{k=0}^i \gamma \circ \tilde{c}_x(t_{\pm k})$$



F . 7.3: Extraction des détails. (a) Les données sont composées d'un champ de directions θ (ici, t_{\min} est affiché en haut en utilisant LIC, avec les singularités sur-imposées en rouge), et d'un champ de hauteurs h ($-\kappa_{\min}$ est affiché en bas, avec des valeurs en niveaux de gris). (b) La trajectoire $\tilde{c}_x(t)$ est réduite par un facteur τ_+ dans le but de stopper le voisinage sur les singularités (en haut); le profil des données p_x est ensuite obtenu via une procédure de *fitting* d'un polynôme cubique \tilde{p}_x (en bas). (c) Les paramètres du profil, comme la distance d_s au squelette (en haut) ou la hauteur de la fonction $\tilde{p}_x(t_x)$ (en bas) sont cohérents spatialement et temporellement.

Nous réduisons ensuite le voisinage de sorte qu'aucune singularité ne soit dépassée pendant l'échantillonnage ; nous identifions la position τ_+ (resp. τ_-) à partir de laquelle Γ_x^+ (resp. Γ_x^-) est plus grand qu'un certain seuil Γ_{\max} (nous utilisons $\Gamma_{\max} = 1$), comme illustré en haut de la Figure 7.3 (b). Le facteur de réduction est choisi comme le minimum de $|\tau_-|$ et $|\tau_+|$. Le voisinage est alors ré-échantillonné avec $2k + 1$ points espacés avec le facteur de réduction le long de la trajectoire, de sorte à avoir assez d'information pour le *fitting* du profil.

7.2.3 Fitting du profil

L'objectif de cette troisième étape d'analyse est d'identifier la position du squelette potentiel le long du voisinage 1D de chacun des pixels, avec des informations complémentaires concernant le profil. Le moyen le plus efficace est de construire une fonction analytique décrivant le champ de hauteurs le long de la trajectoire. Une procédure de *fitting* est ainsi mise en place pour trouver la fonction \tilde{p}_x qui représente au mieux les données mesurées aux t_i le long de $\tilde{c}_x(t)$. Cette technique permet non seulement de caractériser le profil des détails qui sera nécessaire lors de la stylisation des lignes, mais régularise aussi le bruit pouvant être présent dans les données. En pratique, nous souhaitons minimiser la fonction d'énergie suivante :

$$E(\mathbf{x}) = \sum_{i=-k}^k (h \circ \tilde{c}_x(t_i) - \tilde{p}_x(t_i))^2$$

La fonction analytique que nous avons choisie pour représenter les données est un polynôme cubique car il contient juste assez de degrés de libertés pour identifier les extrema autour de chacun des pixels :

$$\tilde{p}_x(t) = at^3 + bt^2 + ct + d$$

Pour calculer les coefficients de cette fonction, nous utilisons une approche basée sur les moindres carrés avec $n = 2k + 1$ échantillons le long de la trajectoire de chaque pixel. Étant donné les n positions t_i , nous construisons la matrice $n \times 4$ nommée \mathbf{X} et composée des lignes $(t_i^3 \ t_i^2 \ t_i \ 1)$. Ces n échantillons déterminent l'ensemble des équations $\mathbf{X}\mathbf{A} = \mathbf{H}$, où \mathbf{A} est le vecteur colonne 4×1 composé des coefficients a, b, c, d à déterminer, et \mathbf{H} est le $n \times 1$ vecteur des valeurs de hauteurs $h(\tilde{c}_x(t_i))$. Nous résolvons cette équation via les moindres carrés : $\mathbf{A} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{H}$. Comme nous utilisons une paramétrisation locale, la matrice $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ peut facilement être calculée, puis multipliée avec la matrice \mathbf{H} pour obtenir les coefficients de la fonction cubique.

Le profil ainsi mesuré pour chaque point \mathbf{x} représente la forme du champ de hauteurs le long de la trajectoire et permet d'identifier les propriétés de la surface avec une expression analytique simple. Le profil montre généralement deux extrema, exprimés par $t_{\alpha,\beta} = (-b \pm \sqrt{b^2 - 3ac})/3a$. La position t_x du squelette est aussi facilement obtenue en choisissant l'extrema pour lequel la dérivée seconde

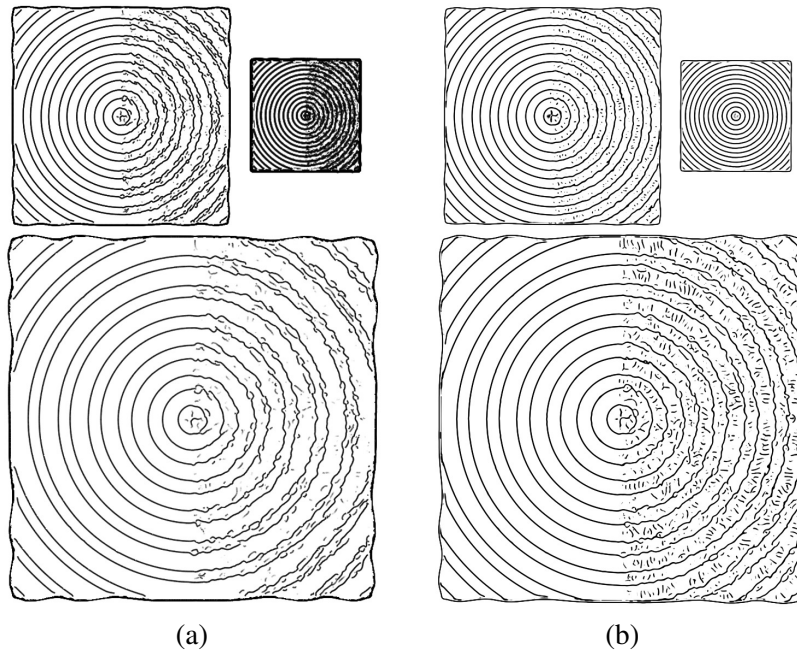


Fig. 7.4: (a) Les détails extraits en espace objet engendrent des agglomérations de lignes lorsque l'objet s'éloigne du point de vue (les crêtes apparentes, ainsi que les vallées ici). (b) Notre formulation des crêtes et des vallées en espace image n'a pas ce problème.

$d^2 \tilde{p}_x(t)/dt^2 = 6at + 2b$ est positive (nous ne le prenons pas en compte lorsqu'un seul minima est trouvé). La hauteur et la courbure du profil sont donnés par $\tilde{p}_x(t_x)$ et $d^2 \tilde{p}_x(t_x)/dt^2$ (car $d\tilde{p}_x(t_x)/dt = 0$).

La Figure 7.3 (c) montre le résultat du processus de *fitting* : la distance de chacun des pixels au squelette le plus proche ($d_s = \|\mathbf{x} - \tilde{c}_x(t_x)\|$) est illustrée avec des gradients de couleurs, et la hauteur du profil $\tilde{p}_x(t_x)$ est dessinée en niveaux de gris. On peut observer que ces deux informations sont cohérentes le long des profils des détails ; cela illustre la cohérence spatiale du processus de *fitting*. La Figure 7.4 compare le squelette des détails extrait avec notre méthode avec les lignes obtenues via la technique des crêtes apparentes [JDA07], à différentes échelles. Du bruit a été ajouté sur la moitié de la fonction d'ondulation ; cela provoque une agglomération dans le cas des crêtes apparentes. Avec notre approche formulée directement en espace image, les lignes obtenues sont propres et le bruit disparaît pour des vues distantes.

Stylisation de la forme via des brosses implicites

Maintenant que nous avons clairement identifié le message que nous souhaitons communiquer (i.e. le profil de chacun des détails avec sa courbure ainsi que la position et la hauteur du squelette pour chacun des pixels), le second challenge consiste à styliser ces détails en assurant une cohérence temporelle. Comme nous l'avons vu dans le Chapitre 6, la plupart des techniques précédentes adoptent une approche basée sur la paramétrisation des lignes, le long desquelles il est possible d'appliquer une fonction ou une texture. Néanmoins, les détails qui sont ainsi projetés en espace image sont sujets à de nombreuses distorsions, étirements, agglomérations ou séparations durant l'animation et provoquent des incohérences temporelles dans le rendu des animations.

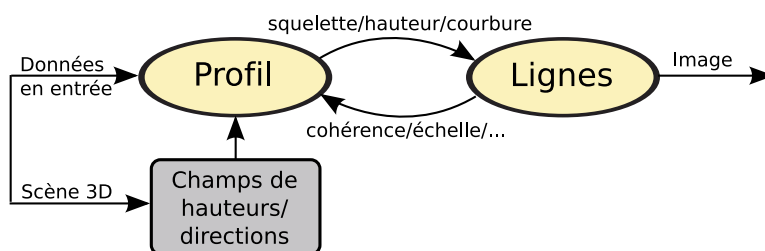


Fig. 8.1: Le message que nous souhaitons communiquer contient les informations du profil (distance au squelette, hauteur, courbure) qui ont été calculées depuis des champs de hauteurs et de directions. Le style consiste à appliquer une brosse le long des profils détectés.

Nous proposons une alternative avec laquelle il n'y a pas besoin de paramétrisation. Elle est basée sur une approche implicite qui permet de convoluer une brosse le long des squelettes, en prenant le profil en compte. Les résultats sont compa-

rables à ceux que l'on pourrait obtenir avec des logiciels comme Photoshop ou Gimp. De plus, nous enrichissons cette méthode de convolution avec l'utilisation de textures cohérentes inspirées des techniques de rendus aquarelles [BKTS06]. Le choix de cette méthode engendre des avantages et des inconvénients sur la gamme de stylisation pouvant être créée. Contrairement aux stylisations de lignes explicites, nous ne possédons qu'une paramétrisation locale du champ de hauteurs que nous avons en entrée ; il nous est donc impossible de plaquer des textures le long des squelettes. Plus généralement, ce choix de brosses implicites ne permet pas de générer tous les styles pouvant être créés avec une définition explicite des lignes. Cette contrainte est en fait nécessaire pour assurer une cohérence temporelle dans les scènes animées. En effet, travailler avec des définitions implicites permet de styliser les différents effets qui peuvent survenir, sans avoir à les suivre image par image : les déformations, les étirements/compressions, les agglomérations/séparations sont ainsi gérés automatiquement dans notre système, sans que des problèmes de cohérence apparaissent. Mais s'il nous faut limiter l'intervalle de styles pouvant être créé pour éviter les problèmes de cohérence temporelles, l'utilisation des Brosses Implicites, combinée aux informations contenues dans le profil, permet aussi de créer des nouveaux styles qui ne pouvaient pas être générés avec les approches précédentes. Nous pouvons représenter ces deux approches de stylisations avec deux ensembles qui s'intersectent. Notre contribution consiste donc à utiliser une approche basée sur une convolution qui imite le contact d'une brosse sur un support en prenant en compte les informations extraites du profil de chacun des pixels. Nous montrons qu'une implémentation de cette technique en temps réel sur le GPU, combinée avec plusieurs brosses et effets de textures, permet de produire une grande variété de styles.

8.1 Convolution basée sur les détails de surface

Intuitivement, notre technique de stylisation consiste à appliquer des empreintes de brosses avec des tailles, des orientations, des couleurs et des opacités variées sur les points qui sont assez proches d'un squelette du champ de hauteurs, et dont le profil est assez important (nous n'appliquons pas la brosse sur les zones quasiment planaires). Les lignes stylisées qui émergent de ce procédé héritent donc de la cohérence spatiale et temporelle des détails extraits depuis la surface originale. Cela peut être vu comme une adaptation des Surfaces de Convolution [BS91] pour communiquer la forme de la surface des objets. Formellement, une Brosse Implicite est une fonction $\mathcal{I} : \mathbb{R}^2 \rightarrow [0, \infty[$ qui permet de faire un mappage entre le point d'une image et une couleur. Elle est définie comme la convolution d'une fonction de poids $w_p : \mathbb{R}^2 \rightarrow [0, 1]$ et d'une fonction de brosse $b_p : \mathbb{R}^2 \rightarrow [0, 1]^4$, toutes deux prenant en compte les propriétés du profil pour chacun des pixels de l'image :

$$\mathcal{I}(\mathbf{y}) = \int_{\mathbb{R}^2} w_p(\mathbf{x}) b_p(\mathbf{y} - \mathbf{x}) d\mathbf{x} \quad (8.1)$$

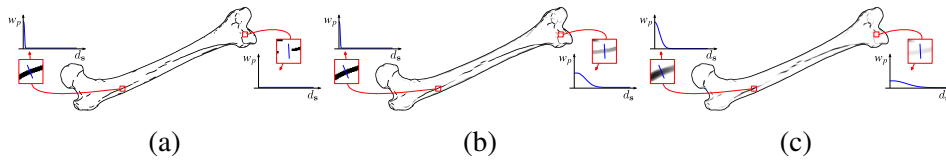


FIG. 8.2: Fonctions de poids. (a) Avec $\lambda_p = 1$ et $\sigma_p = \epsilon$, seules des lignes binaires sont sélectionnées. (b) Avec λ_p proportionnel à la hauteur du profil, les lignes disparaissent progressivement avec les détails. (c) Avec σ_p proportionnel à la courbure du profil, les lignes s'étalent autour des détails lisses.

Sur chacun des points de l'image, \mathcal{I} mesure l'accumulation des contributions des empreintes pondérées sur les canaux RVBA. Nous transformons ces valeurs pour les réarranger dans l'intervalle $[0, 1]^4$ via une mise à l'échelle homogène sur les quatre canaux, comme ce qui est fait classiquement dans les techniques d'adaptation de ton [Sch94b]. La différence principale entre les Brosses Implicites et les Surfaces de Convolution [BS91] est que les fonctions définissant la brosse dépendent du profil des détails (i.e. du message calculé dans le chapitre précédent) ; c'est le sens de l'indice p utilisé dans ce chapitre.

La fonction de poids permet de contrôler implicitement quels sont les points de l'image qui sont proches d'un squelette et dont le profil est assez saillant. Elle est définie sur un point \mathbf{x} par :

$$w_p(\mathbf{x}) = \lambda_p \exp^{-d_s^2/2\sigma_p^2} \quad (8.2)$$

où d_s est, comme précédemment, la distance au squelette le plus proche, et λ_p et σ_p sont les paramètres basés sur le profil de la surface et définissent respectivement le pic de la fonction gaussienne et son écart type. Les choix de λ_p et σ_p peuvent être très variés. Nous montrons trois combinaisons typiques dans la Figure 8.2, avec une brosse représentée par un petit disque. Avec $\lambda_p = 1$ et $\sigma_p = \epsilon$ (a), seules des lignes binaires très fines sont détectées (comme dans un rendu pouvant être obtenu avec des stylos encre). En choisissant λ_p proportionnel à la hauteur du profil (b), les lignes disparaissent doucement lorsque les détails s'estompent (comme avec un crayon). Avec σ_p proportionnel à la courbure du profil (c), les lignes s'étendent et s'élargissent autour des détails assez lisses (comme avec un aérographe). Nous laissons à l'utilisateur le choix d'expérimenter les nombreuses autres combinaisons possibles pour λ_p et σ_p .

La fonction de brosse permet, quant à elle, de contrôler comment l'empreinte de la brosse (sélectionnée par l'utilisateur) est positionnée, orientée et mise à l'échelle en prenant en compte les détails de surface. Nous séparons l'empreinte de sa position pour fournir un meilleur contrôle à l'utilisateur :

$$b_p(\mathbf{u}) = f_p \circ T_p(\mathbf{u}) \quad (8.3)$$

où f_p est la fonction qui permet de déterminer la couleur de l’empreinte définie dans son propre espace paramétrique, et T_p une transformation des points de l’image dans cet espace paramétrique. Dans notre système, nous utilisons une similitude pour T_p . Le fait que ces deux fonctions dépendent des propriétés des détails permet de corrélérer le style avec les caractéristiques du profil, comme on peut le voir dans la Figure 8.3 : T_p permet par exemple d’orienter l’empreinte le long du squelette (a). Dans (b), T_p applique une mise à l’échelle sur l’empreinte, de manière proportionnelle à la hauteur du profil. f_p permet par exemple d’assigner la couleur de l’intensité lumineuse sur la position du squelette (c). D’autres combinaisons pourraient être trouvées pour T_p et f_p dans le but d’obtenir des styles différents et originaux.

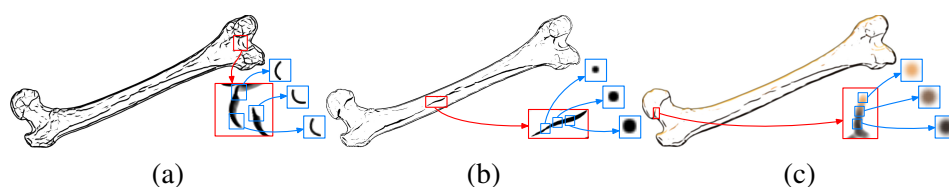


FIG. 8.3: En variant f_p et T_p , nous obtenons différents styles corrélés aux propriétés du profil. (a) T_p oriente l’empreinte de la brosse le long du squelette. (b) T_p met l’empreinte à l’échelle en prenant en compte la hauteur du profil. (c) f_p prend sa couleur depuis le rendu original de l’objet 3D à la position du squelette.

Les lignes qui sont créées avec cette approche sont fortement corrélées aux détails de la surface grâce au choix des fonctions de poids et de brosse. Cependant, pour des raisons esthétiques et pour augmenter la variété de styles, il est aussi important d’incorporer des variations qui ne sont pas forcément corrélées aux propriétés de la surface. On peut citer par exemple l’addition d’une texture de papier en arrière plan, des lignes interrompues (comme des pointillés), des variations de couleurs ou même des effets de déformations des lignes pour les faire vaciller. Dans notre système, ces effets sont obtenus en modulant chacun des composants de l’équation 8.1 avec une texture de bruit additionnelle, notée η , qui va permettre de les perturber d’une manière similaire aux techniques de rendus aquarelles [BNTS07, BCGF10]. L’effet papier est obtenu en multipliant directement \mathcal{I} par η . Les interruptions de lignes sont produites en multipliant λ_p par η . Les variations de couleurs sont obtenues en modulant la couleur de f_p et les effets d’ondulation sont créés en modulant les paramètres de T_p avec η .

8.2 Applications

Le prototype que nous avons mis en place fournit une interface de conception de brosses inspirée de celles que l'on peut trouver dans les logiciels de traitement d'images bitmap comme Photoshop ou Gimp. Quelques exemples de styles et de brosses sont donnés dans la Figure 8.4 : l'utilisateur choisit une texture de brosse ou une fonction (a), ajuste facilement les paramètres du poids et de l'empreinte (b), et peut ajouter des perturbations dues au bruit (c-d) en observant les variations de styles en temps-réel.

Nous utilisons des textures de bruit pour produire les effets de papiers, d'interruption des lignes, de couleurs ou d'ondulation. Néanmoins, le fait d'aligner des textures sur l'écran peut produire des effets de glissements indésirables (appelé effet "rideau de douche") qui apparaissent souvent dans les rendus de type aquarelles. Nous utilisons alors des solutions récentes que nous incorporons dans notre système. Elles permettent d'éviter ces problèmes, notamment en combinant des textures fractales à des échelles multiples [BCGF10].

L'avantage principal de notre technique de convolution par rapport aux solutions précédentes plus complexes est qu'elle est entièrement *dynamique* : le style est entièrement contrôlé par le choix des fonctions de poids et de brosse, et l'algorithme fonctionne en temps-réel, sans précalculs tout en assurant une cohérence temporelle. Cette méthode est donc parfaitement adaptée à une implémentation sur

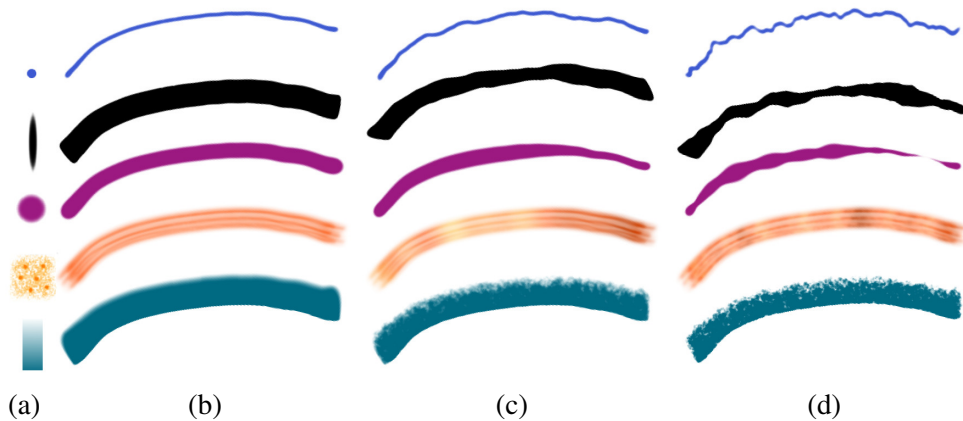


Fig. 8.4: Quelques exemples de styles. Chacune des lignes montre (a) une empreinte dans son espace paramétrique, (b) appliquée sur une courbe simple, (c) avec une petite perturbation ou (d), avec une perturbation plus prononcée. De haut en bas, nous perturbons la position, l'orientation, la taille et la couleur de la brosse et nous ajoutons une texture de papier.



Fig. 8.5: Rendu d'une stèle gravée de Saint-Émilien. Les symboles et les détails de cette pierre correspondent tous deux à des vallées, mais avec différents profils. Nous communiquons cette distinction en variant la taille de la brosse en fonction de la courbure du profil.

le GPU. Notre prototype fonctionne sur une carte NVidia G-480, avec une définition de ligne affichée à un moment donné. En pratique, nous calculons d'abord les valeurs de poids et les paramètres de transformation de la brosse par pixel dans un *fragment shader*, en utilisant les données extraites du profil en entrée. L'image des Brosses Implicites est ensuite obtenue en appliquant un carré texturé par pixel, combiné à un mélange additif des couleurs. L'opacité de chacun des carrés est simplement obtenue en prenant la valeur de poids sur son pixel correspondant. Les rotations, translations et mises à l'échelles du carré sont mises en œuvre en prenant en compte les paramètres de transformation de la brosse. Enfin le carré est rempli avec l'empreinte créée dans une texture ou avec une fonction procédurale simple. Un opérateur d'adaptation de ton est appliqué dans une dernière passe.

Notre technique de stylisation vise plusieurs types d'applications. Tout d'abord, elle est bien adaptée aux illustrations scientifiques. Les lignes stylisées sont étroitement corrélées aux propriétés de la surface, comme le montre la Figure 8.5, où les nombreuses vallées de cette pierre gravée sont communiquées avec des lignes dont la largeur dépend de la courbure du profil.

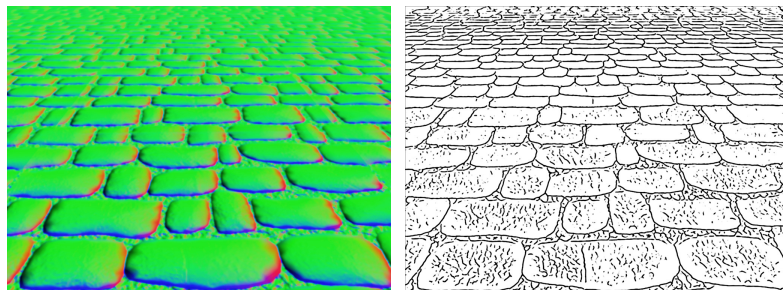


Fig. 8.6: Notre méthode fonctionne naturellement avec des objets composés de cartes de normales. Les vallées sont extraites ici.

La méthode fonctionne aussi parfaitement avec les phénomènes dynamiques comme les fluides et pourrait être facilement implémentée dans des jeux vidéo. En effet, elle communique naturellement les détails trouvés sur les cartes de normales (Figure 8.6) et fonctionne avec des objets déformables (Figure 8.7).

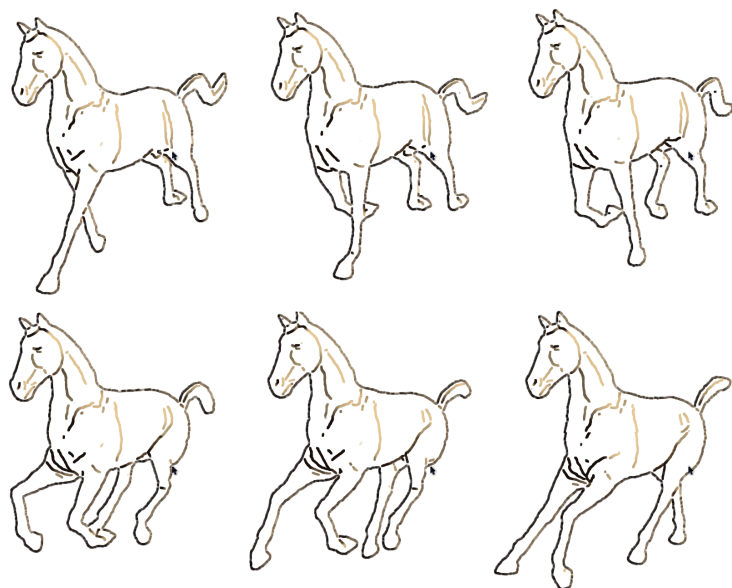


Figure 8.7: Exemple d'un modèle 3D déformable. Les lignes calculées sur ce modèle de cheval animé sont stylisées avec les couleurs du rendu original.

Les Brosses Implicites peuvent aussi être utilisées pour la stylisation de vidéos. Si les cartes de normales et de profondeurs sont disponibles (e.g. exportées depuis une application de rendu 3D), alors les multiples détails peuvent être communiqués dans une image, en appliquant notre technique sur chacun des types de détail et en les assemblant par la suite. Ce procédé est illustré dans la Figure 8.8, qui montre la séquence vidéo d'un modèle de terrain avec crêtes, vallées, et silhouettes rendues dans des styles différents. De la même manière, il est possible d'appliquer la technique sur une vidéo standard, comme on peut le voir sur l'exemple d'une goutte d'eau dans la Figure 8.9, où les lignes stylisées sont dessinées seules ou sur-imposées sur les images originales. Dans ce cas, nous utilisons les arêtes de luminance et nous prenons seulement les singularités de directions en compte, car les occultations ne sont pas disponibles.

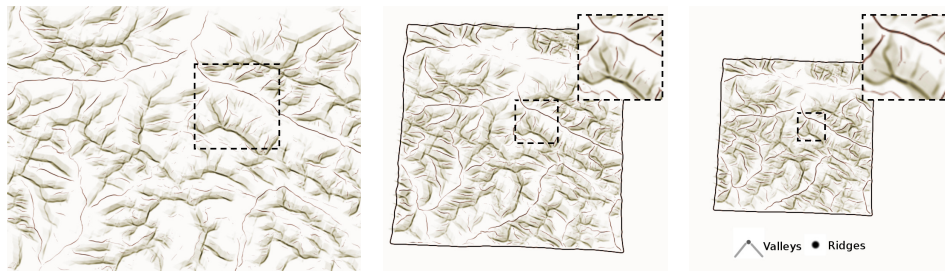


FIG. 8.8: Ce terrain 3D est rendu avec deux styles différents pour les crêtes et les vallées. Les détails se simplifient automatiquement à des échelles différentes, et les stylisations correspondantes se mélangent correctement.

8.3 Comparaison des stylisations

Nous comparons seulement avec les méthodes qui s'adaptent aussi à des phénomènes de distorsions pouvant survenir dans les animations. Les techniques fonctionnant avec des lignes paramétrées (avec une définition explicite) représentent un moyen intuitif pour les styliser : une texture est appliquée le long du squelette et la stylisation des jonctions ou des fins de lignes (les *end-points*) peut facilement être contrôlée (avec par exemple l'effacement progressif des lignes, ou une forme d'empreinte différente en ces lieux). En contraste, notre approche permet de décorréler la texture du squelette, et les lignes stylisées s'arrêtent automatiquement sur les *end-points* via la fonction de poids. Même si cela permet un intervalle de styles très grand, le contrôle de l'utilisateur est moins direct avec les Brosses Implicites : comme dans les logiciels comme Photoshop ou Gimp, la forme de la brosse qui a été choisie ne correspond pas directement au résultat obtenu. C'est néanmoins nécessaire pour éviter les problèmes de cohérence temporelle qui peuvent survenir durant la séparation ou l'agglomération de lignes. D'un autre côté, notre formulation implicite permet de créer des styles qui dépendent de la forme de la surface

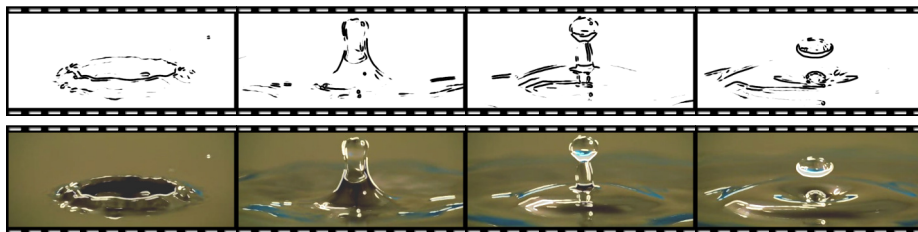


FIG. 8.9: Les images d'une vidéo stylisée, où les arêtes de luminosité sont communiquées avec des lignes dont la largeur varie. Elles sont dessinées en noir (en haut) ou en blanc sur l'image originale (en bas) avec une brosse qui a la forme d'un disque.

étudiée, comme l'étalement des lignes lorsque la surface est assez lisse, et pourrait s'adapter très facilement dans un système de composition basé sur des nœuds, comme Nuke¹ ou Fusion².

La méthode de Lee *et al.* [LMLH07] partage certains avantages avec les Brosses Implicites, mais possède aussi d'importantes limitations. Leur technique ne permet de communiquer que les détails qui correspondent à des changements d'intensité de lumière. Ils emploient une méthode de *fitting* 2D itératif qui nécessite plusieurs étapes pour converger. De plus, ils ne gèrent pas les occultations et obtiennent donc des résultats qui ne sont pas toujours cohérents spatialement. Leur technique de stylisation aussi est limitée : elle ne permet de contrôler que l'épaisseur des lignes et des problèmes de glissements surviennent lorsqu'ils combinent le rendu avec des textures.

8.4 Discussion

La méthode implicite appliquée à la fois à l'extraction et à la stylisation des détails que nous avons mise en œuvre permet de créer des rendus au traits cohérents spatialement et temporellement sur des scènes 3D dynamiques en temps-réel. Les performances de notre système dépendent de la résolution utilisée, du nombre d'itérations lors de la diffusion anisotrope et de la taille de la brosse. Par exemple, nous obtenons 82 *fps* à une résolution de 1024×768 et une empreinte de 10 pixels. Les performances chutent lorsqu'on augmente le nombre d'itérations : nous obtenons 56, 40 et 31 *fps* pour respectivement 10, 20 et 30 itérations.

Chacune des étapes de notre système pourrait aussi être utilisée séparément. Par exemple, les détails extraits dynamiquement pourraient être stylisés avec une approche explicite en les suivant et en les paramétrant au préalable. Néanmoins, il n'existe pas à ce jour de techniques (encore moins en temps-réel) capables de gérer les problèmes de distorsions sans introduire de phénomènes de scintillements ou de glissements dans le rendu. Cela pourrait être un domaine de recherche intéressant à étudier, mais cette solution pourrait nécessiter un processus complexe, impliquant d'inspecter les images futures pour suivre les événements de distorsion potentiels qui pourraient survenir. À l'inverse, nous pourrions aussi appliquer notre procédé de convolution pour styliser le squelette des lignes extraites directement à la surface des maillages (en espace objet), ou sur des courbes dessinées à la main. Cependant, nous ne pourrions pas adapter le style des lignes aux propriétés du profil dans ce cas, car ces informations ne seraient pas disponibles.

Concernant les détails que nous pouvons détecter, la limitation principale de notre approche se situe sur les singularités. Nous les prenons actuellement en compte

¹<http://www.thefoundry.co.uk/>

²<http://eyeonline.com>

en stoppant la trajectoire en ces lieux. Mais le nombre de singularités augmente avec l'ordre des définitions de lignes (cf. Figure 7.3 (a)). De plus, notre méthode ignore certains types de détails comme les jonctions, précisément parce que ces points correspondent à des singularités. Il serait donc intéressant de trouver une nouvelle définition des détails qui limite le nombre de points singuliers, tout en donnant la possibilité d'identifier facilement les jonctions. De plus, nous pourrions extraire d'autres propriétés de la surface, en plus du profil de chaque squelette. Par exemple, la courbure du squelette lui même pourrait s'avérer très utile pour styliser différemment (mais toujours implicitement) les traits droits des traits courbés.

Concernant la stylisation des lignes, une limitation importante se situe sur les *end-points* qui sont automatiquement corrélés aux détails qui deviennent trop lisses. Une solution intéressante serait de distribuer des particules le long des lignes, comme dans les travaux de Vanderhaeghe *et al.* [VBTS07]. Elles pourraient être utilisées pour interagir avec la fonction de poids et créer par exemple des *end-points* additionnels. Nous souhaitons examiner cette solution de stylisation à base de particules dans le futur. Elle pourrait notamment permettre d'obtenir des résultats qui se rapprochent de rendus imitant les peintures.

Les lignes stylisées que nous obtenons avec l'approche des Brosses Implicites montrent des résultats cohérents, même sur des scènes dynamiques qui peuvent contenir des objets qui bougent en arrière plan. Cependant, le résultat obtenu pourrait ne pas être celui attendu pour des objets structurés comme les grilles ou les vagues de l'eau. Dans ce cas, il est possible que le résultat attendu dépende d'une connaissance plus sémantique concernant la structure des objets. Une approche fonctionnant en espace image ne serait alors plus suffisante ; il faudrait étudier et simplifier les objets de manière sémantique pour que la forme finale stylisée des lignes corresponde aux attentes de l'utilisateur.

Troisième partie

Discussions et conclusion

Vers un nouveau modèle d'apparence

Nous avons vu deux nouvelles approches pour communiquer la forme dans le rendu de scènes 3D. Dans la première, nous modifions directement l'équation de réflexion pour mettre en valeur les détails quels que soient l'environnement lumineux et le matériau utilisé. Dans la seconde, nous introduisons une méthode générale pour détecter et styliser des lignes qui correspondent à des définitions différentes de détails. Ces deux approches manipulent les données physiques (maillages, environnements, matériaux) en entrée pour créer le message qui permet ensuite d'être réintroduit dans plusieurs styles. Nous avons pu voir aussi comment le style influe sur le message qui doit être mis en place. Suivant la manière dont on souhaite mettre les détails en valeur et les primitives choisies pour le rendu, le message doit être adapté pour pouvoir être intégré dans des styles spécifiques. Une information de courbure dépendante du point de vue était par exemple nécessaire pour modifier les rayons lumineux qui interviennent dans l'équation du rendu. Pour le rendu au trait, une information plus précise concernant la forme de la surface a du être prise en compte. Dans ces deux approches, le contrôle sur l'apparence du rendu final reste limité.

9.1 Observations

Différentes observations dans les travaux que nous avons présentés montrent à quel point la manipulation des données physiques est délicate et peut entraîner des contraintes et des effets indésirables dans les rendus finaux.

9.1.1 Forme, lumière, matériau

Dans le Chapitre 5, nous modifions la fréquence ou l'intensité des motifs lumineux reflétés sur la surface d'objets 3D. Sans toucher à la géométrie des objets (position des sommets et normales), nous permettons de mettre en valeur les détails

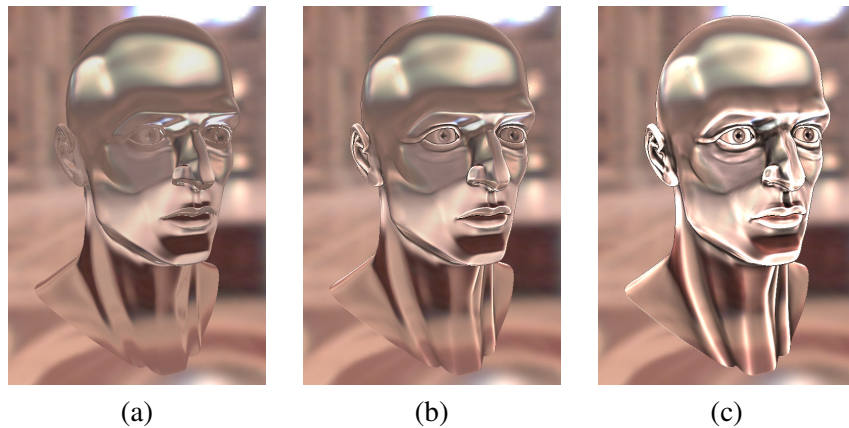


Fig. 9.1: Application de la technique de mise à l'échelle des intensités lumineuse avec un matériau parfaitement réfractif. (a) Rendu original. (b) L'intensité de la mise en valeur permet de mettre en valeur les détails sans trop perturber le matériau. (c) L'amplitude du signal est tellement modifiée que le matériau semble complètement dénaturé.

quels que soient les *BRDFs* et les styles utilisés. L'un des principaux inconvénients est que ces méthodes modifient aussi notre perception des matériaux. En modifiant localement la lumière sur les points des objets, nous influons donc à la fois sur la perception de la forme et des matériaux utilisés. La Figure 9.1 montre qu'avec la technique de modification de la radiance sur un objet parfaitement réfractif, il est possible de dénaturer fortement le matériau original. Cet exemple montre qu'en modifiant un paramètre physique (ici la lumière) nous agissons sur la perception des autres éléments qui interviennent : la forme et le matériau. D'autres travaux permettent de montrer ces effets en jouant sur chacun des paramètres. Dans la Figure 9.2 (a), Fleming et Caniard [CF07] montrent par exemple que la forme d'un même objet composé d'un matériau Lambertien peut être perçue différemment suivant la manière dont on positionne la source de lumière. Dans la Figure 9.2 (b), seule la forme change entre les deux images. Pourtant, Vangorp *et al.* [VLD07] montrent que celle-ci influe largement sur notre perception des matériaux. Des effets équivalents interviennent en modifiant non pas la forme mais l'environnement (cf. Figure 9.2 (c)). Nous pouvons déduire de tous ces travaux que la modification, même légère, d'un élément physique peut entraîner des perceptions très différentes de ces mêmes éléments : la forme des surfaces, l'environnement utilisé et les matériaux sont des propriétés intimement liées lors de la production d'une image de synthèse, et il est très difficile (voire impossible) de prévoir directement quel sera le résultat de la modification de l'une ou l'autre des données physiques en entrée. Il serait donc intéressant de manipuler ces données différemment dans le but de faciliter la création de l'apparence des objets dans les images. En jouant sur des paramètres plus pertinents pour la perception et en permettant de modifier le rendu de manière cohérente, les artistes auraient la possibilité de créer des images beaucoup plus facilement en anticipant les résultats sur le plan perceptuel.

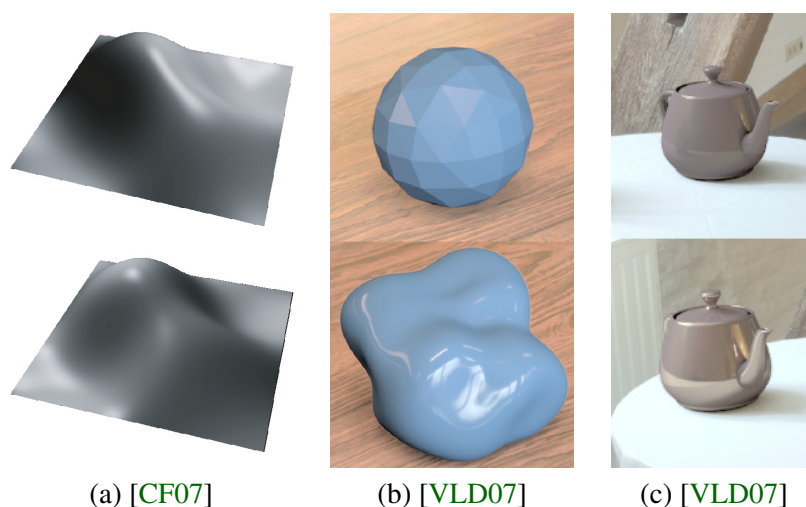
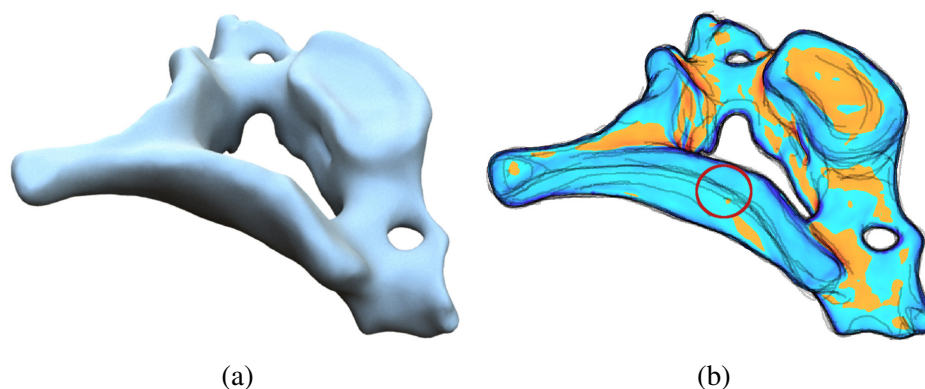


FIG. 9.2: Influence de la modification de paramètres physiques sur les autres paramètres. (a) Même forme, même matériau ; une légère modification de la direction lumineuse influence notre perception de la forme. (b) Même environnement lumineux, même matériaux ; des objets de forme différentes peuvent influencer notre perception des matériaux. (c) Même forme, Même matériaux ; la modification de l'environnement lumineux entraîne des différences sur notre perception des matériaux.

9.1.2 Variations de forme VS variations d'intensités lumineuses

Même dans les dessins au trait, massivement utilisés dans les applications scientifiques pour représenter la forme des objets, il n'existe pas de règles précises pour représenter toutes les données pertinentes. Dans le Chapitre 5, nous réintroduisons les détails en modifiant l'équation de réflectance pour éviter les effets de masquage qui peuvent apparaître dans les rendus originaux. Mais l'effet inverse peut survenir : certains détails peuvent être exagérés dans les rendus classiques, de telle sorte que la perception de la forme soit dénaturée. Dans l'étude de Cole *et al.* [CGL⁺08], il est demandé à des artistes de faire des dessins au trait à partir de rendus originaux dans l'objectif de représenter au mieux la forme des objets. Nous avons vu dans le Chapitre 7 que les artistes ne se limitent pas à une seule définition de lignes et communiquent les occlusions, les crêtes, les vallées, mais aussi les points d'inflexion suivant l'échelle que l'on souhaite représenter. Dans l'exemple de la Figure 9.3, nous avons sur-imposé les lignes dessinées par les artistes sur notre descripteur. On remarque que non seulement, les lignes correspondent à des détails de surface différents comme ceux que l'on a évoqués précédemment, mais aussi à d'autres caractéristiques comme on peut le voir dans le cercle rouge : les lignes dessinées sur cette zone ne correspondent pas à une propriété importante de la surface (la courbure ne permet pas de détecter des détails dans cette région).

En examinant le rendu initial, on constate alors que les artistes ont perçu des variations d'intensité correspondant plutôt à des changements d'environnement et de matériau comme étant dues à la forme. Cet exemple montre que les variations qui interviennent dans le rendu peuvent être perçues comme des changements de forme ou de matériau et de lumière. L'une des questions essentielles et très intéressante à étudier pour la manipulation de l'apparence est de comprendre quelles sont les interactions entre forme, lumière et matière qui vont donner lieu à la perception de variations dues à la forme ou à l'éclairage.



F . 9.3: (a) Rendu original à partir duquel les artistes devaient représenter la forme avec des lignes. (b) Les lignes des artistes sont accumulées et superposées sur notre descripteur de forme local pour montrer les correspondances entre la forme et le rendu original.

9.2 Besoins d'un nouveau modèle d'apparence

Comme nous l'avons constaté, l'interaction entre ces différents éléments n'est pas facile à appréhender et les variations créées peuvent être perçues comme étant dues soit à la forme, soit aux matériaux et à l'environnement. Il serait donc intéressant de manipuler des données qui ont un sens au niveau perceptuel pour contrôler la manière dont nous percevons les variations dans le rendu final des images. Un tel jeu de données permettrait de manipuler facilement l'apparence mais aussi de connaître la manière dont elle se comporte si on modifie les paramètres.

Les peintres, les illustrateurs scientifiques ou même les photographes ont déjà une très bonne intuition sur la manière de procéder suivant qu'il s'agisse par exemple de mettre en valeur la forme ou les matériaux dans une scène. Dans l'exemple de la Figure 9.4 (a), Hogarth montre qu'il est possible de manipuler le rendu en plaçant les lumières de telle sorte que la forme apparaisse le mieux possible. Ce type de rendu, appelé *Sculptural Lighting*, est composé d'un matériau proche du Lambertien, quoi qu'un peu brillant. Les lumières sont disposées de manière à éclairer

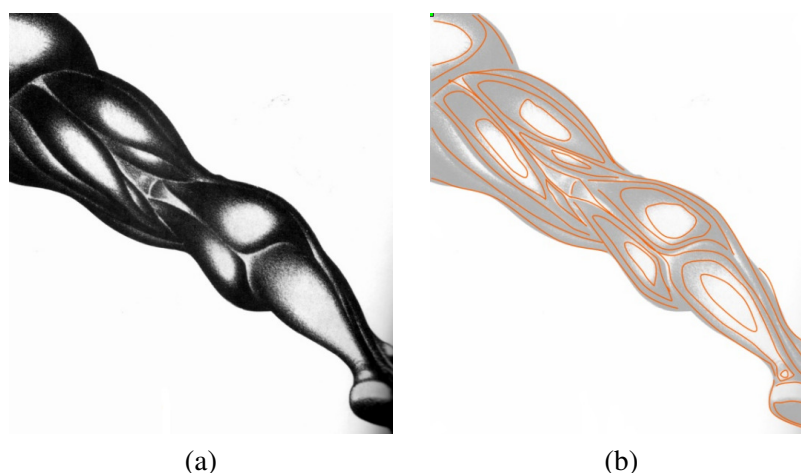


Figure 9.4: (a) Lumière sculpturale de Hogarth. La forme est représentée de manière idéale. (b) Les isolignes montrent que les variations suivent parfaitement les formes de la jambe.

chacune des parties de la jambe séparément. Comme on peut le voir dans la Figure 9.4 (b) dans laquelle nous avons tracé les isolignes de la luminance, toutes les variations qui apparaissent sont essentiellement dues à la forme. A l’opposé, les dessins d’art-déco contiennent souvent des formes très simples. La Figure 9.5 (a) montre un exemple de Tamara De Lempicka dans lequel nous pouvons voir la représentation d’une femme. Dans le zoom (cf. Figure 9.5 (b)), on peut voir que les formes sont limitées à des formes géométriques simples comme des sphères et des ondulations. Les variations qui apparaissent sont alors essentiellement dues aux propriétés des matériaux. La couleur et les variations qui apparaissent sur les sphères nous permettent de deviner aisément qu’il s’agit de raisins au matériau translucide. Les cheveux aussi possèdent des isolignes qui ne suivent pas exactement les formes ondulées dont ils sont composés.

La recherche d’un nouveau modèle d’apparence ayant la particularité de contrôler la manière dont nous percevons les variations pourrait permettre de résoudre les limitations dont souffrent les approches que nous avons présentées dans ce manuscrit. En effet, un tel modèle pourrait permettre de savoir quelles sont les zones qui sont cachées ou qui sont déjà suffisamment mises en valeur dans les rendus originaux. L’exagération des détails pourrait alors se faire seulement dans les régions de surface qui en ont besoin, sans perturber la perception des matériaux utilisés. Dans le cas du rendu au trait, nous pourrions déterminer facilement les lieux et les profils qui correspondent à des zones de variation intéressantes en ayant la possibilité de prendre ou non en compte la lumière et les matériaux, et en contournant les problèmes dus aux singularités.

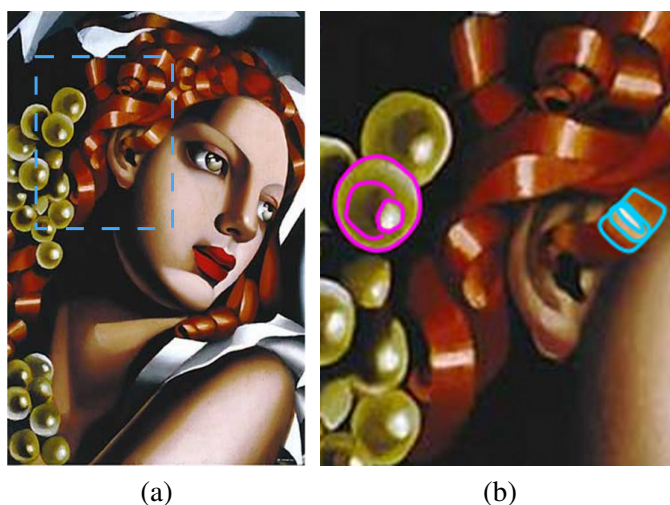


Fig. 9.5: (a) Exemple de dessin art-déco de Tamara De Lempicka. (b) Sur le raisin et dans les cheveux, les formes sont très simples et les variations sont essentiellement dues aux matériaux.

La Figure 9.6 illustre la manière dont un modèle d'apparence pourrait fonctionner. En analysant les données physiques en entrée, il devrait être possible d'extraire des données pertinentes au niveau perceptuel. En manipulant des nouvelles données, il devrait être plus facile de contrôler les variations perçues qui sont plutôt dues à la forme ou aux matériaux. Celles-ci pourraient alors être utilisées pour obtenir des descripteurs plus pertinents qui constitueraient une base très intéressante pour modéliser les informations nécessaires au message. Ce modèle pourrait donc constituer une base pour détecter facilement les informations utiles. De plus, l'intervalle des styles pouvant être créés ne serait pas limité à la mise en valeur de la forme à travers le rendu ou avec des rendus au trait : nous pourrions exploiter ce modèle pour étudier des styles tels que les hachures ou des rendus plus abstraits comme l'aquarelle, qui ont la capacité de communiquer efficacement les propriétés des matériaux.



Fig. 9.6: Un modèle d'apparence basé sur des données perceptuelles pourrait permettre de contrôler la manière dont la forme ou les matériaux sont perçus dans les images créées.

9.3 Deux angles d'attaque

Bien entendu, il nous faut trouver les propriétés qui caractérisent au mieux les informations pertinentes au niveau perceptuel. Un travail en étroite collaboration avec les chercheurs en perception doit être réalisé pour obtenir des solutions correctes. Nous proposons deux études qui pourraient permettre de progresser dans le développement d'un tel système.

9.3.1 Étude de l'équation de réflexion

Tous les effets qui interviennent dans le rendu d'une image sont décrits par l'équation de réflexion 3.1 présentée dans le Chapitre 3. En analysant les variations de cette fonction non pas à la surface des objets comme dans [RMB07], mais dans l'image, nous pourrions déterminer les relations complexes qui interviennent entre la lumière, la forme et le matériau. On peut en effet raisonnablement penser qu'en calculant la dérivée de cette fonction en espace image, des relations étroites entre ces différents éléments apparaîtront. Nous pourrions alors étudier analytiquement à quoi sont dues les variations qui apparaissent dans le rendu en faisant des choix et des hypothèses sur certains éléments. En faisant par exemple l'hypothèse d'une surface simple comme une sphère et d'un environnement lumineux contenant peu de variations, nous pourrions déterminer sur quelles zones les variations changent lorsque l'on modifie le matériau.

L'objectif d'une telle analyse serait donc de mieux comprendre l'origine des variations qui interviennent dans le rendu d'une scène. Plus précisément, cette étude pourrait nous permettre de déterminer l'environnement qui maximise les variations qui sont dues à la forme dans le rendu final, avec un matériau donné. Il s'agit d'un problème ouvert avec des applications qui s'avèrent aussi très intéressantes. Quels sont, par exemple, les types et dispositions de lumière à utiliser dans un musée pour révéler au mieux la forme des objets exposés ? La même approche pourrait aussi servir à mettre en valeur non pas la forme, mais les matériaux utilisés. Le défi est encore plus grand ici, car il n'existe pas à ce jour d'études qui établissent exhaustivement quels sont les détails pertinents qui interviennent dans la perception des matériaux.

9.3.2 Étude d'éléments perceptuels

Une autre manière d'aborder le problème est d'étudier des caractéristiques pertinentes qui interviennent dans le système visuel humain. Ces propriétés pourraient aussi intervenir directement dans le modèle d'apparence car elles sont déterminantes dans notre perception de la forme. Les silhouettes sont par exemple une caractéristique essentielle pour déterminer la forme globale des objets qui composent une scène. Il est clair qu'elles ont une grande influence sur notre perception de la forme [FTA04] et elles devront donc être prises en compte dans un modèle

d'apparence. Elles permettent en effet d'identifier les différents objets dans une scène.

Les détails de surface perçus pourraient être décrits par un flot de directions. Dans le Chapitre 7, nous avons vu que les profils détectés le long des trajectoires permettaient de montrer les détails les plus pertinents de la surface. Il a aussi été montré que les directions de courbures principales jouent un rôle déterminant dans notre perception de la forme des objets. Dans l'étude de Fleming et Holtmann-Rice [FHR10], les auteurs montrent qu'en convoluant un bruit le long d'un champ de directions 2D correspondant à des courbures principales, une perception 3D des objets apparaît aux utilisateurs (cf. Figure 9.7). Un flot de directions pourrait donc être une bonne représentation de la forme. Néanmoins, le flot décrit dans [FHR10], ou ceux utilisés pour détecter les lignes dans les chapitres précédents ne sont pas suffisants. En effet, les champs de directions que nous avons extraits sont composés de nombreuses singularités. Par exemple, les directions de courbure principales ne sont pas définies sur les points d'inflexion. De la même manière, les directions de variation de courbures ne sont pas définies sur les crêtes et les vallées. Plus précisément, les lieux où les directions ne sont pas définies pour les dérivées d'ordre pairs seront ceux où elles seront bien définies pour les ordres impairs. L'association de deux ordres semble alors nécessaire pour obtenir un flot de directions cohérent à la surface des objets. Une base de départ pour l'étude d'un flot cohérent pourrait être d'utiliser des filtres [FA91] qui permettent de détecter des lignes avec l'utilisation de deux ordres successifs ; mais la technique devra être améliorée pour gérer des caractéristiques importantes telles que les jonctions.

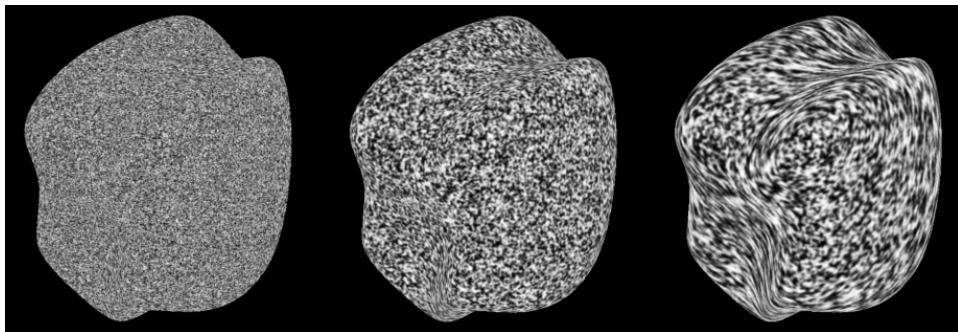


Figure 9.7: En convoluant un bruit (avec la méthode des *line Integral Convolution* [CL93] le long des directions de courbures principales, la forme de l'objet apparaît.

Enfin, des motifs représentant l'interaction entre la lumière et les matériaux pourraient être appliqués le long du flot pour obtenir le rendu final. Ces primitives pourraient alors être capables de représenter différents types de matériaux : diffus, brillants ou même transparents et translucides.

Tous ces éléments pourraient être intégrés dans un nouveau modèle pour contrôler l'apparence des rendus d'objets. D'autres caractéristiques dont nous n'avons pas parlé pourraient bien sûr être prises en compte. Les niveaux de détail, le mouvement des primitives, la cohérence spatiale sont autant d'éléments à prendre en compte pour obtenir un modèle pertinent. Un modèle comme celui-ci pourrait non seulement permettre de modéliser des scènes 3D avec des propriétés perceptuellement plausibles, mais aussi d'être utilisé dans les applications de dessin vectoriel, ou pour décomposer les images ou les vidéos. Nous pourrions par exemple imaginer imiter le rendu d'une scène 3D sans données physiques en entrées ; en manipulant seulement les données perceptuelles, il serait peut être possible de reproduire l'apparence de scènes 3D. A l'opposé, ce modèle pourrait être utile pour aider à la décomposition, à l'analyse et au traitement des images ou des vidéos.

Conclusion

10.1 Résumé des contributions

Tout au long de ce manuscrit, nous avons pu constater que les relations entre *messages* et *styles* sont très étroites. Dans notre cas, le message consiste à exagérer la forme, et doit s'adapter aux contraintes de styles pour fournir des paramètres cohérents. Nous nous sommes tout d'abord intéressés à la manière de réintroduire la forme au travers de l'éclairage. Les travaux précédents étant limités en termes de message et de style, nous avons décidé de modifier directement l'équation de réflexion pour mettre en valeur les détails de surface des objets 3D dans des styles variés. Deux étapes ont été nécessaires pour mettre cette technique en œuvre.

Le message doit tout d'abord être extrait à partir des informations géométriques de la surface des objets. Pour cela, nous avons mis en place une méthode permettant de calculer en temps-réel un tenseur de courbure en utilisant les seules cartes de normales et de profondeurs. Ce descripteur local permet de représenter des informations pertinentes sur la surface des objets avec de nombreux avantages par rapport aux informations de courbures qui étaient extraites en espace objet dans les approches précédentes : il est entièrement dépendant du point de vue, les niveaux de détail sont automatiques et contrôlables et il est parfaitement adapté à des scènes dynamiques car il est calculé en temps-réel et ne nécessite pas de précalcul.

Ce descripteur a ensuite été intégré dans l'équation de réflexion de trois manières différentes. Dans la première, l'intensité et le signe de la courbure sont directement utilisés lors du rendu pour éclaircir, adoucir ou même inverser les couleurs dans des styles cartoon ou minimaux. En modifiant dans un deuxième temps la direction de la radiance entrante, nous avons montré qu'il était possible d'exagérer la forme des objets dans des rendus qui restent plausibles, sans que la perception des matériaux soit trop altérée. La dernière modification que nous avons proposée consistait à ajouter une fonction de mise à l'échelle prenant en compte à la fois les propriétés

de surface et de matériaux directement dans l'équation de réflexion. Cette solution, qui consiste à modifier l'amplitude du signal plutôt que sa fréquence, a l'avantage d'être flexible et rapide à mettre en œuvre dans tout type d'application, ludiques ou scientifiques.

Dans la seconde partie du manuscrit, nous nous sommes intéressés au dessin au trait stylisé. Nous avons montré que les travaux précédents se limitent généralement à la définition des lignes ou à leurs stylisations. De plus, aucune des méthodes ne permet de styliser des lignes en gardant une cohérence spatiale et temporelle lors des animations. Les lignes sont soit extraites explicitement (en espace objet), puis stylisées avec des techniques de suivi de paramétrisation qui ne sont pas complètement cohérentes, soit extraites implicitement (en espace image) et cohérentes mais très limitées en termes de stylisation.

Nous avons donc mis en place une méthode implicite générale permettant d'extraire la plupart des définitions de lignes existantes comme les occultations, les crêtes, les vallées ou les inflexions en prenant en compte le point de vue. Cette technique, qui ne prend en entrée qu'un champ de hauteurs et un champ de directions, fonctionne en temps-réel sur les cartes graphiques récentes tout en permettant de gérer automatiquement les niveaux de détail. De plus, nous détectons le profil de chacun des détails pour permettre une grande variété de styles basés sur la forme de la surface.

La stylisation de ces lignes est obtenue en convoluant une brosse dont la forme, la position, l'orientation ou les couleurs correspondent aux informations contenues dans le profil des détails. Comme cette stylisation est elle-même implicite (tout comme l'extraction des détails), ce processus est cohérent spatialement et temporellement. Cette méthode est donc parfaitement adaptée aux scènes dynamiques. Nous avons aussi offert à l'utilisateur une grande variété de choix possibles dans la stylisation. Grâce aux informations du profil et aux autres effets pouvant être ajoutés comme l'utilisation des textures de bruits cohérentes, il est possible d'obtenir des styles qui n'avaient pas été implémentés jusqu'à maintenant.

10.2 Nouveaux messages / nouveaux styles

Notre objectif principal est de donner la possibilité à l'utilisateur de contrôler l'apparence d'une scène, avec un message et un style qui lui est propre. Comme nous l'avons vu dans le Chapitre 9, nous souhaitons analyser les données physiques (forme/environnement/matériau) en entrée pour créer un nouveau modèle d'apparence capable de contrôler la manière dont nous percevons les variations de forme et de matériaux dans les rendus. Ce modèle d'apparence pourrait être une base à partir de laquelle il serait beaucoup plus facile d'extraire les informations nécessaires à un message et à un style donné (cf. Figure 10.1). Comme nous

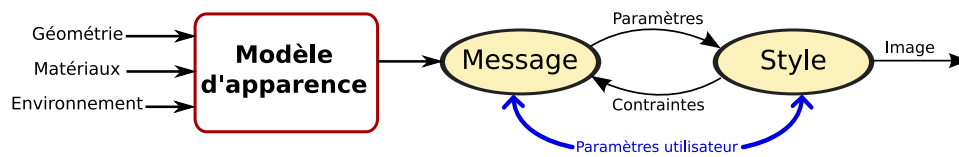


Fig. 10.1: Le modèle d'apparence pourrait générer des informations intuitives à manipuler, qui prennent en compte non seulement la forme, mais aussi l'environnement et les matériaux. L'extraction des paramètres nécessaires au message serait plus simple et les styles plus variés.

L'avons déjà évoqué précédemment, le message pourrait consister à mettre en avant d'autres informations, comme les matériaux ou les relations spatiales entre les objets. L'utilisation d'un unique descripteur de forme ne paraît en effet pas suffisant pour éviter, par exemple, de modifier la perception des matériaux dans les rendus lorsque l'on souhaite modifier l'équation de réflexion.

D'un autre côté, nous souhaitons aussi être capable d'appliquer une plus grande variété de styles. Des rendus impliquant des hachures, des coups de pinceaux ou encore de l'aquarelle, représentent un défi dans le domaine du rendu expressif, car il est très difficile d'obtenir des résultats plausibles et cohérents spatialement et temporellement. Dans le cas d'une peinture, les coups de pinceaux doivent suivre des directions privilégiées à la surface des objets, mais doivent aussi bouger de manière cohérente lors d'un changement de topologie ou de point de vue. Nous pensons que l'étude du nouveau modèle d'apparence pourrait s'avérer utile car il permettrait notamment d'obtenir un champ de directions cohérent à la surface des objets.

Publications

Publications avec comité de lecture

- [VPB⁺11] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christophe Schlick. Improving Shape Depiction under Arbitrary Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2011. À paraître.
- [VPB⁺10] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christophe Schlick. Radiance Scaling for Versatile Surface Enhancement. *I3D '10 : Proc. symposium on Interactive 3D graphics and games*, 2010. Best papers award.
- [VPB⁺09] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christophe Schlick. Light warping for enhanced surface depiction. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2009.
- [VBGS08a] Romain Vergne, Pascal Barla, Xavier Granier, and Christophe Schlick. Apparent relief : a shape descriptor for stylized shading. *NPAR '08 : Proc. international symposium on Non-photorealistic animation and rendering*, 2008.

Présentation

- [VBGS08a] Romain Vergne, Pascal Barla, Xavier Granier, and Christophe Schlick. Shading with Apparent Relief. *SIGGRAPH '08 : ACM SIGGRAPH 2008 talks*, 2008.

Publications sans comité de lecture

- [VVJ⁺10] Romain Vergne, David Vanderhaeghe, Chen Jiazhou, Pascal Barla, Xavier Granier, and Christophe Schlick. Implicit Brushes for Stylized Line-based Rendering. Research Report, INRIA, 2010. En cours de soumission.
- [VVBB10] David Vanderhaeghe, Romain Vergne, Pascal Barla, and William Baxter. Bidirectional Appearance Distribution Function for Stylized Shading. Research Report, INRIA, 2010. Travaux en cours.
- [VBT⁺07] Romain Vergne, Adrien Bousseau, Joëlle Thollot, David Vanderhaeghe, Pascal Barla, and Xavier Granier. Utilisation du rendu expressif pour l'illustration et l'exploration de données archéologiques. *Virtual Retrospect 2007 : Archéologie et Réalité Virtuelle.*, 2007.
- [VG07] Romain Vergne, and Xavier Granier. Stylisation d'objets éclairés par des cartes d'environnement HDR. *Actes des 20èmes Journées de l'Association Française d'Informatique Graphique (AFIG)*, 2007.

Dérivation du jacobien

Le but de cette annexe est de dériver la formulation du jacobien introduite au chapitre 5. Le jacobien J est défini par $J = \frac{d\mathbf{l}}{d\mathbf{l}'}$ et il intervient dès qu'un changement de variable est effectué dans un calcul d'intégrales.

Dans le repère local $\{\mathbf{u}, \mathbf{v}, \mathbf{z}\}$ où \mathbf{u} et \mathbf{v} sont les directions des courbures principales exprimées en espace image (la caméra pointant vers $-\mathbf{z}$), on note les directions \mathbf{l} et \mathbf{l}' de la lumière de la façon suivante :

$$\mathbf{l} = \begin{pmatrix} l_u \\ l_v \\ l_z \end{pmatrix} = \begin{pmatrix} \cos(\phi) \sin(\theta) \\ \sin(\phi) \sin(\theta) \\ \cos(\theta) \end{pmatrix} \quad (\text{A.1})$$

$$\mathbf{l}' = \begin{pmatrix} l'_u \\ l'_v \\ l'_z \end{pmatrix} = \begin{pmatrix} \cos(\phi') \sin(\theta') \\ \sin(\phi') \sin(\theta') \\ \cos(\theta') \end{pmatrix} \quad (\text{A.2})$$

Dans cette paramétrisation $d\mathbf{l} = \sin \theta d\theta d\phi$. En introduisant $\tau = \tan(\theta/2)$, on remarque que

$$d\tau = 2/(1 + \tau^2)d\theta$$

et que

$$\sin(\theta) = 2\tau/(1 + \tau^2).$$

Ce qui conduit à :

$$d\mathbf{l} = \frac{4\tau}{(1 + \tau^2)^2} d\tau d\phi.$$

La projection stéréographique introduit deux nouvelles variables :

$$a = l_u/(1 + l_z) = \cos(\phi)\tau$$

et

$$b = l_v/(1 + l_z) = \sin(\phi)\tau.$$

Puisque $a^2 + b^2 = \tau^2$ et que $da db = \tau d\tau d\phi$, cette nouvelle substitution nous permet d'obtenir :

$$dl = \frac{4}{(1 + a^2 + b^2)^2} da db. \quad (\text{A.3})$$

Dans l'espace stéréographique, le warping de la direction correspond simplement à une mise à l'échelle de a et b : $a' = \lambda_u a$ et $b' = \lambda_v b$. Ce qui donne :

$$dl = \frac{4}{(1 + a'^2/\lambda_u^2 + b'^2/\lambda_v^2)^2} \frac{da'}{\lambda_u} \frac{db'}{\lambda_v} \quad (\text{A.4})$$

$$= \frac{4\lambda_u\lambda_v}{(\lambda_u^2\lambda_v^2 + \lambda_v^2a'^2 + \lambda_u^2b'^2)^2} da' db'. \quad (\text{A.5})$$

À l'instar de l'équation A.3, on peut écrire :

$$dl' = \frac{4}{(1 + a'^2 + b'^2)^2} da' db'. \quad (\text{A.6})$$

En combinant ce résultat avec l'équation A.5 on obtient alors :

$$dl = \frac{\lambda_u^3\lambda_v^3(1 + a'^2 + b'^2)^2}{(\lambda_u^2\lambda_v^2 + \lambda_v^2a'^2 + \lambda_u^2b'^2)^2} dl'. \quad (\text{A.7})$$

a' et b' correspondent à la projection stéréographique de la direction \mathbf{l}' , $a' = l'_u/(1 + l'_z)$ et $b' = l'_v/(1 + l'_z)$ et on obtient donc :

$$dl = \frac{\lambda_u^3\lambda_v^3((1 + l'_z)^2 + l'^2_u + l'^2_v)^2}{(\lambda_u^2\lambda_v^2(1 + l'_z)^2 + \lambda_v^2l'^2_u + \lambda_u^2l'^2_v)^2} dl' = J dl'. \quad (\text{A.8})$$

Comme \mathbf{l}' est une direction on a : $1 - l'^2_z = l'^2_u + l'^2_v$, ce qui permet d'obtenir :

$$J = \frac{4\lambda_u^3\lambda_v^3(1 + l'_z)^2}{(\lambda_u^2\lambda_v^2(1 + l'_z)^2 + \lambda_v^2l'^2_u + \lambda_u^2l'^2_v)^2}.$$

Bibliographie

- [ACSD⁺03] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3) :485–493, 2003.
- [Ade01] E. H. Adelson. On seeing stuff : the perception of materials by humans and machines. In B. E. Rogowitz and T. N. Pappas, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4299 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 1–12, June 2001.
- [AH03] Ken-ichi Anjyo and Katsuaki Hiramitsu. Stylized highlights for cartoon rendering and animation. *IEEE Comput. Graph. Appl.*, 23(4) :54–61, 2003.
- [APS00] Michael Ashikhmin, Simon Premoze, and Peter Shirley. A microfacet-based BRDF generator. In *Proc. ACM SIGGRAPH '00*, pages 65–74. ACM, 2000.
- [AS00] Michael Ashikhmin and Peter Shirley. An anisotropic phong brdf model. *J. Graph. Tools*, 5(2) :25–32, 2000.
- [AVBSZ07] Y. Adato, Y. Vasilyev, O. Ben Shohar, and T. Zickler. Toward a theory of shape from specular flow. In *ICCV07*, pages 1–8, 2007.
- [BBT⁺06] Pascal Barla, Simon Breslav, Joëlle Thollot, François Sillion, and Lee Markosian. Stroke pattern analysis and synthesis. In *Computer Graphics Forum (Proc. of Eurographics 2006)*, volume 25, 2006.
- [BBT09] Pierre Bénéard, Adrien Bousseau, and Joëlle Thollot. Dynamic solid textures for real-time coherent stylization. In *I3D '09 : Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 121–127, New York, NY, USA, 2009. ACM.
- [BCGF10] Pierre Bénéard, Forrester Cole, Aleksey Golovinskiy, and Adam Finkelstein. Self-similar texture for coherent line stylization. In *NPAR 2010 : Proceedings of the 8th International Symposium on*

- Non-photorealistic Animation and Rendering*, pages 91–97. ACM Press, 2010.
- [BG06] Stefan Bruckner and M. Eduard Groller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5) :1077–1084, 2006.
- [BG07] Stefan Bruckner and Meister Eduard Gröller. Style transfer functions for illustrative volume rendering. *Computer Graphics Forum*, 26(3) :715–724, September 2007.
- [BG10] William Baxter and Naga Govindaraju. Simple data-driven modeling of brushes. In *I3D '10 : Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 135–142, New York, NY, USA, 2010. ACM.
- [BKTS06] Adrien Bousseau, Matthew Kaplan, Joëlle Thollot, and François Sillion. Interactive watercolor rendering with temporal coherence and abstraction. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*. ACM, 2006.
- [BL04] William V. Baxter and Ming C. Lin. A versatile interactive 3d brush model. In *PG '04 : Proceedings of the Computer Graphics and Applications, 12th Pacific Conference*, pages 319–328, Washington, DC, USA, 2004. IEEE Computer Society.
- [Bli77] James F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77 : Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, New York, NY, USA, 1977. ACM.
- [BLV⁺10] Pierre Bénéard, Ares Lagae, Peter Vangorp, Sylvain Lefebvre, George Drettakis, and Joëlle Thollot. A dynamic noise primitive for coherent stylization. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering 2010)*, 29(4) :1497–1506, june 2010.
- [BNTS07] Adrien Bousseau, Fabrice Neyret, Joëlle Thollot, and David Salesin. Video watercolorization using bidirectional texture advection. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3), 2007.
- [BPK⁺07] Mario Botsch, Mark Pauly, Leif Kobbelt, Pierre Alliez, Bruno Lévy, Stephan Bischoff, and Christian Rössl. Geometric modeling based on polygonal meshes. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 courses*, page 1, New York, NY, USA, 2007. ACM.
- [BRO08] Jonathan Bronson, Penny Rheingans, and Marc Olano. Semi-automatic stencil creation through error minimization. In *NPAR '08 : Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 31–37, New York, NY, USA, 2008. ACM.

- [BRS05] Tamy Boubekeur, Patrick Reuter, and Christophe Schlick. Scalar tagged pn triangles. In *EUROGRAPHICS 2005 (Short Papers)*. Eurographics, 2005.
- [BS91] Jules Bloomenthal and Ken Shoemake. Convolution surfaces. *SIGGRAPH Comput. Graph.*, 25(4) :251–256, 1991.
- [BS05] Tamy Boubekeur and Christophe Schlick. Generic mesh refinement on gpu. In *ACM SIGGRAPH/Eurographics Graphics Hardware*, 2005.
- [BSD08] Louis Bavoil, Miguel Sainz, and Rouslan Dimitrov. Image-space horizon-based ambient occlusion. In *SIGGRAPH '08 : ACM SIGGRAPH 2008 talks*, pages 1–1, New York, NY, USA, 2008. ACM.
- [BSD09] Michael Balzer, Thomas Schlömer, and Oliver Deussen. Capacity-constrained point distributions : A variant of Lloyd’s method. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2009)*, 28(3) :86 :1–8, 2009.
- [BSK04] Mario Botsch, Michael Spornat, and Leif Kobbelt. Phong splatting. *Symposium on Point-Based Graphics*, pages 25–32, 2004.
- [BSSB99] P. Brunet, R. Scopigno, Mario Costa Sousa, and John W. Buchananz. Computer-generated graphite pencil rendering of 3d polygonal models. In *Computer Graphics Forum - Proceedings of Eurographics '99*, 1999.
- [BSSS07] John Brosz, Faramarz F. Samavati, M. T. Carpendale Sheelagh, and Mario Costa Sousa. Single camera flexible projection. In *NPAR '07 : Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 33–42, New York, NY, USA, 2007. ACM.
- [BT75] Phong Bui-Tuong. Illumination for computer generated pictures. In *CACM*, pages 311–317, 1975.
- [BTBP07] Alexandra Baer, Christian Tietjen, Ragnar Bade, and Bernhard Preim. Hardware-accelerated stippling of surfaces derived from medical volume data. In *Eurographics/IEEE-VGTC Symposium on Visualization*, pages 235–242, 2007.
- [BTM06] Pascal Barla, Joëlle Thollot, and Lee Markosian. X-toon : An extended toon shader. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*. ACM, 2006.
- [BTS05] Pascal Barla, Joëlle Thollot, and François X. Sillion. Geometric clustering for line drawing simplification. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Sketches*, page 96, New York, NY, USA, 2005. ACM.
- [BW07] Harlen Costa Batagelo and Shin-Ting Wu. Estimating curvatures and their derivatives on meshes of arbitrary topology from sampling directions. *Vis. Comput.*, 23(9) :803–812, 2007.

- [BWL04] William Baxter, Jeremy Wendt, and Ming C. Lin. Impasto : a realistic, interactive model for paint. In *NPAR '04 : Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pages 45–148, New York, NY, USA, 2004. ACM.
- [BZOP07] Simon Bouvier-Zappa, Victor Ostromoukhov, and Pierre Poulin. Motion cues for illustration of skeletal motion capture data. In *Non-Photorealistic Animation and Rendering 2007*, pages –, August 2007.
- [Can86] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6) :679–698, 1986.
- [CAO09] Jianghao Chang, Benoît Alain, and Victor Ostromoukhov. Structure-aware error diffusion. In *SIGGRAPH Asia '09 : ACM SIGGRAPH Asia 2009 papers*, pages 1–8, New York, NY, USA, 2009. ACM.
- [CAS⁺97] Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin. Computer-generated watercolor. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on computer graphics and interactive techniques*, pages 421–430, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [CBWG10] Nelson Chu, William Baxter, Li-Yi Wei, and Naga Govindaraju. Detail-preserving paint modeling for 3d brushes. In *NPAR '10 : Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pages 27–34, New York, NY, USA, 2010. ACM.
- [CDH06] Liviu Coconu, Oliver Deussen, and Hans-Christian Hege. Real-time pen-and-ink illustration of landscapes. In *NPAR '06 : Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 27–35, New York, NY, USA, 2006. ACM.
- [CF07] Franck Caniard and Roland W. Fleming. Distortion in 3D shape estimation with changes in illumination. In *APGV '07 : Proc. symposium on Applied perception in graphics and visualization*, pages 99–105. ACM, 2007.
- [CF08] Forrester Cole and Adam Finkelstein. Partial visibility for stylized lines. In *NPAR 2008*, June 2008.
- [CF09] Forrester Cole and Adam Finkelstein. Fast high-quality line visibility. In *Proceedings of I3D 2009*, pages 115–120, February 2009.
- [CGL⁺08] Forrester Cole, Aleksey Golovinskiy, Alex Limpaecher, Heather Stoddart Barros, Adam Finkelstein, Thomas Funkhouser, and Szymon Rusinkiewicz. Where Do People Draw Lines ? *ACM Trans. Graph. (Proc. SIGGRAPH 2008)*, 27(3) :1–11, 2008.
- [CH02] J. P. Collomosse and P. M. Hall. Painterly rendering using image salience. In *EGUK '02 : Proceedings of the 20th UK conference on Eurographics*, page 122, Washington, DC, USA, 2002. IEEE Computer Society.

- [CH03] John P. Collomosse and Peter M. Hall. Cubist style rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics*, 9(4) :443–453, 2003.
- [CJG09] Gregory Cipriano, George N. Phillips Jr., and Michael Gleicher. Multi-scale surface descriptors. *IEEE Trans. Vis. Comput. Graph.*, 15(6) :1201–1208, 2009.
- [CL93] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. In *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 263–270, New York, NY, USA, 1993. ACM.
- [CM99] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *ICCV '99 : Proceedings of the International Conference on Computer Vision-Volume 2*, page 1197, Washington, DC, USA, 1999. IEEE Computer Society.
- [Cor70] Tom Cornsweet. *Visual Perception*. New York : Academic Press, 1970.
- [CP03] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *SGP '03 : Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 177–187, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [CP09] A.N.M. Imroz Choudhury and Steven G. Parker. Ray tracing npr-style feature lines. In *NPAR '09 : Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, pages 5–14, New York, NY, USA, 2009. ACM.
- [CS04] Patrick Coleman and Karan Singh. Ryan : rendering your animation nonlinearly projected. In *NPAR '04 : Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pages 129–156, New York, NY, USA, 2004. ACM.
- [CSD⁺09] Forrester Cole, Kevin Sanik, Doug DeCarlo, Adam Finkelstein, Thomas Funkhouser, Szymon Rusinkiewicz, and Manish Singh. How well do line drawings depict shape ? In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, volume 28, August 2009.
- [CSM03] David Cohen-Steiner and Jean-Marie Morvan. Restricted delaunay triangulations and normal cycle. In *SCG '03 : Proceedings of the nineteenth annual symposium on Computational geometry*, pages 312–321, New York, NY, USA, 2003. ACM.
- [CST05] Paolo Cignoni, Roberto Scopigno, and Marco Tarini. A simple Normal Enhancement technique for Interactive Non-photorealistic Renderings. *Comp. & Graph.*, 29(1) :125–133, 2005.
- [CT05] Nelson S.-H. Chu and Chiew-Lan Tai. Moxi : real-time ink dispersion in absorbent paper. *ACM Trans. Graph.*, 24(3) :504–511, 2005.

- [CTP⁺03] Matthieu Cunzi, Joëlle Thollot, Sylvain Paris, Gilles Debunne, Jean-Dominique Gascuel, and Frédo Durand. Dynamic canvas for immersive non-photorealistic walkthroughs. In *Proc. Graphics Interface*. A K Peters, LTD., june 2003.
- [DBB06] Philip Dutré, Kavita Bala, and Philippe Bekaert. *Advanced Global Illumination (Second Edition)*. A. K. Peters, Ltd., 2006.
- [dC76] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [DCFR07] Christopher DeCoro, Forrester Cole, Adam Finkelstein, and Szymon Rusinkiewicz. Stylized shadows. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, August 2007.
- [DCoY03] Iddo Dror, Daniel Cohen-or, and Hezy Yeshurun. Example-based style synthesis. In *In Proceedings of Computer Vision and Pattern Recognition (Madison, USA)*, pages 143–150, 2003.
- [DFR04] Doug DeCarlo, Adam Finkelstein, and Szymon Rusinkiewicz. Interactive rendering of suggestive contours with temporal coherence. In *NPAR '04 : Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pages 15–145, New York, NY, USA, 2004. ACM.
- [DFRS03] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive Contours for Conveying Shape. *ACM Trans. Graph. (Proc. SIGGRAPH 2003)*, 22(3) :848–855, 2003.
- [DHvOS00] Oliver Deussen, Stefan Hiller, Cornelius van Overveld, and Thomas Strothotte. Floating points : A method for computing stipple drawings. *Computer Graphics Forum*, 19 :40–51, 2000.
- [DR07] Doug DeCarlo and Szymon Rusinkiewicz. Highlight lines for conveying shape. In *NPAR '07 : Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 63–70, New York, NY, USA, 2007. ACM.
- [DR08] Julie Dorsey and Holly Rushmeier. Light and materials in virtual cities. In *SIGGRAPH 2008 Course Notes*, 2008.
- [DS02] Doug DeCarlo and Anthony Santella. Stylization and abstraction of photographs. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 769–776, New York, NY, USA, 2002. ACM.
- [DSSL98] Julie Dorsey, Arthur C. Smith, Jonathan Levene, and Jonathan Levene. A framework for non-realistic projections. Technical report, Master of Engineering Thesis, MIT, 1998.
- [Dur02] Frédo Durand. An invitation to discuss computer depiction. In *NPAR '02 : Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 111–124, New York, NY, USA, 2002. ACM.

- [DvGNK99] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1) :1–34, 1999.
- [EWS08] Elmar Eisemann, Holger Winnemöller, John C. Hart, and David Salesin. Stylized vector art from 3d models with region support. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering 2008)*, 27(4), June 2008.
- [FA91] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 :891–906, 1991.
- [FDA03] Roland W. Fleming, Ron O. Dror, and Edward H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision*, pages 347–368, 2003.
- [FHR10] Roland Fleming and Daniel Holtmann-Rice. Shape from Smear. *Journal of Vision*, 10(7) :74, 2010.
- [Fie87] David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 4 :2379–2394, 1987.
- [FMS02] Bert Freudenberg, Maic Masuch, and Thomas Strothotte. Real-time halftoning : a primitive for non-photorealistic shading. In *EGRW '02 : Proceedings of the 13th Eurographics workshop on Rendering*, pages 227–232, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
- [FTA04] Roland W. Fleming, Antonio Torralba, and Edward H. Adelson. Specular reflections and the perception of shape. *J. Vis.*, 4(9) :798–820, 9 2004.
- [FTA09] Roland W. Fleming, Antonio Torralba, and Edward H. Adelson. *Three dimensional shape perception*, chapter Shape from sheen. Springer Verlag, 2009. to appear.
- [FTP03] William T. Freeman, Joshua B. Tenenbaum, and Egon C. Pasztor. Learning style translation for the lines of a drawing. *ACM Trans. Graph.*, 22(1) :33–46, 2003.
- [Fun03] Jennifer Fung. Pen-and-ink textures for real-time rendering. In *IN PROC. OF GRAPHICS INTERFACE 2003*, pages 131–138, 2003.
- [GBP06] Gael Guennebaud, Loïc Barthe, and Mathias Paulin. Splat/Mesh Blending, Perspective Rasterization and Transparency for Point-Based Rendering. In *IEEE/EUROGRAPHICS/ACM Symposium on Point-Based Graphics*, pages 49–58. EUROGRAPHICS, 2006.
- [GCS02] Bruce Gooch, Greg Coombe, and Peter Shirley. Artistic vision : painterly rendering using computer vision techniques. In *NPAR*

- '02 : *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 83–ff, New York, NY, USA, 2002. ACM.
- [GGSC98] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A Non-Photorealistic Lighting Model For Automatic Technical Illustration. In *Proc. ACM SIGGRAPH '98*, pages 447–452. ACM, 1998.
- [GI04] Jack Goldfeather and Victoria Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23(1) :45–63, 2004.
- [GSG⁺99] Bruce Gooch, Peter-Pike J. Sloan, Amy Gooch, Peter Shirley, and Richard Riesenfeld. Interactive technical illustration. In *I3D '99 : Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 31–38, New York, NY, USA, 1999. ACM.
- [GTDS04] Stéphane Grabli, Emmanuel Turquin, Frédo Durand, and François Sillion. Programmable style for npr line drawing. In *Rendering Techniques 2004 (Eurographics Symposium on Rendering)*. ACM Press, june 2004.
- [GTDS10] Stéphane Grabli, Emmanuel Turquin, Frédo Durand, and François X. Sillion. Programmable rendering of line drawing from 3d scenes. *ACM Trans. Graph.*, 29(2) :1–20, 2010.
- [GVH07] Todd Goodwin, Ian Vollick, and Aaron Hertzmann. Isophote distance : a shading approach to artistic stroke thickness. In *NPAR '07 : Proc. international symposium on Non-photorealistic animation and rendering*, pages 53–62. ACM, 2007.
- [Hae90] Paul Haeberli. Paint by numbers : abstract image representations. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 207–214, New York, NY, USA, 1990. ACM.
- [Ham93] B. Hamann. *Curvature approximation for triangulated surfaces*, pages 139–153. Number 8. Springer-Verlag, London, UK, 1993.
- [Her98] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460, New York, NY, USA, 1998. ACM.
- [Her01] Aaron Hertzmann. Paint by relaxation. In *CGI '01 : Computer Graphics International 2001*, pages 47–54, Washington, DC, USA, 2001. IEEE Computer Society.
- [Her02] Aaron Hertzmann. Fast paint texture. In *NPAR '02 : Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 91–ff, New York, NY, USA, 2002. ACM.

- [HJO⁺01] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, New York, NY, USA, 2001. ACM.
- [HLM06] Yun-Xian Ho, Michael S. Landy, and Laurence T. Maloney. How direction of illumination affects visually perceived surface roughness. *J. Vis.*, 6(5) :634–648, 5 2006.
- [HLT⁺09] Thomas Hurtut, Pierre-Edouard Landes, Joëlle Thollot, Yann Gousseau, Rémy Drouilhet, and Jean-François Coeurjolly. Appearance-guided synthesis of element arrangements by example. In *NPAP 2009 : Proceedings of the 7th International Symposium on Non-photorealistic Animation and Rendering*. ACM Press, 2009.
- [Hod03] Elaine R. S. Hodges. *The Guild Handbook of Scientific Illustration*. Wiley, 2003.
- [Hog91] Burne Hogarth. *Dynamic Light and Shade*. Watson Guptill, 1991.
- [HP00] Aaron Hertzmann and Ken Perlin. Painterly rendering for video and interaction. In *NPAP '00 : Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 7–12, New York, NY, USA, 2000. ACM.
- [HPW05] Klaus Hildebrandt, Konrad Polthier, and Max Wardetzky. Smooth feature lines on surface meshes. In *SGP '05 : Proceedings of the third Eurographics symposium on Geometry processing*, page 85, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.
- [HS02] Eyal Hameiri and Ilan Shimshoni. Estimating the principal curvatures and the darboux frame from real 3d range data. *3D Data Processing Visualization and Transmission, International Symposium on*, 0 :258, 2002.
- [HTER04] Christopher G. Healey, Laura Tateosian, James T. Enns, and Mark Remple. Perceptually based brush strokes for nonphotorealistic visualization. *ACM Trans. Graph.*, 23(1) :64–96, 2004.
- [HZ00] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 517–526, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [IB06] Tobias Isenberg and Angela Brennecke. Abstract g-strokes : A concept for simplifying line stylization, 2006.
- [IFH⁺03] Tobias Isenberg, Bert Freudenberg, Nick Halper, Stefan Schlechtweg, and Thomas Strothotte. A developer's guide to silhouette algorithms for polygonal models. *IEEE Comput. Graph. Appl.*, 23(4) :28–37, 2003.

- [JDA07] Tilke Judd, Frédo Durand, and Edward H. Adelson. Apparent Ridges for Line Drawing. *ACM Trans. Graph. (Proc. SIGGRAPH 2007)*, 26(3) :19, 2007.
- [JEGPO02] Pierre-Marc Jodoin, Emeric Epstein, Martin Granger-Piché, and Victor Ostromoukhov. Hatching by example : a statistical approach. In *NPAR '02 : Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 29–36, New York, NY, USA, 2002. ACM.
- [JL97] Bruno Jobard and Wilfrid Lefer. Creating evenly-spaced streamlines of arbitrary density. In *Proceedings of the 8th Eurographics Workshop on Visualization in Scientific Computing*, pages 43–56, 1997.
- [JMLH01] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518, New York, NY, USA, 2001. ACM.
- [JNLM05] Kyuman Jeong, Alex Ni, Seungyong Lee, and Lee Markosian. Detail control in line drawings of 3d meshes, 2005.
- [Kaj86] J.T. Kajiya. The rendering equation. In *Proc. SIGGRAPH'86*, pages 143–150, 1986.
- [KDMF03] Robert D. Kalnins, Philip L. Davidson, Lee Markosian, and Adam Finkelstein. Coherent stylized silhouettes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3) :856–861, July 2003.
- [KHG09] Sebastian Knödel, Martin Hachet, and Pascal Guitton. Interactive generation and modification of cutaway illustrations for polygonal models. In *International Symposium on Smart Graphics*, Lecture Notes on Computer Science. Springer, May 2009. To appear.
- [KJ03] Doorn A.J. van Koenderink J.J. *The visual neurosciences*, chapter Shape and shading, pages 1090–1105. MIT Press, Cambridge, 2003.
- [KKD09] Jan Eric Kyprianidis, Henry Kang, and Jürgen Döllner. Image and video abstraction by anisotropic kuwahara filtering. *Computer Graphics Forum*, 28(7) :1955–1963, 2009. Special issue on Pacific Graphics 2009.
- [KKD10] J. E. Kyprianidis, H. Kang, and J. Döllner. Anisotropic kuwahara filtering on the gpu. In W. Engel, editor, *GPU Pro - Advanced Rendering Techniques*. AK Peters, 2010.
- [KL08] Henry Kang and Seungyong Lee. Shape-simplifying image abstraction. *Computer Graphics Forum*, 27(7) :1773–1780, 2008.
- [KLC07] Henry Kang, Seungyong Lee, and Charles K. Chui. Coherent line drawing. In *NPAR '07 : Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 43–50, New York, NY, USA, 2007. ACM.

- [KLC09] Henry Kang, Seungyong Lee, and Charles K. Chui. Flow-based image abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 15(1) :62–76, 2009.
- [KMM⁺02] Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. WYSIWYG NPR : Drawing strokes directly on 3D models. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 21(3) :755–762, July 2002.
- [KMN⁺99] Michael A. Kowalski, Lee Markosian, J. D. Northrup, Lubomir Bourdev, Ronen Barzel, Loring S. Holden, and John F. Hughes. Art-based rendering of fur, grass, and trees. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 433–438, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [KNS⁺09] Evangelos Kalogerakis, Derek Nowrouzezahrai, Patricio Simari, James Mccrae, Aaron Hertzmann, and Karan Singh. Data-driven curvature for real-time line drawing of dynamic scenes. *ACM Trans. Graph.*, 28(1) :1–13, 2009.
- [Koe90] Jan J. Koenderink. *Solid shape*. MIT Press, Cambridge, MA, USA, 1990.
- [KSFC02] Allison W. Klein, Peter-Pike J. Sloan, Adam Finkelstein, and Michael F. Cohen. Stylized video cubes. In *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 15–22, New York, NY, USA, 2002. ACM.
- [KSL⁺08] Dongyeon Kim, Minjung Son, Yunjin Lee, Henry Kang, and Seungyong Lee. Feature-guided image stippling. *Computer Graphics Forum*, 27(4) :1209–1216, 2008.
- [KST08] Michael Kolomenkin, Ilan Shimshoni, and Ayellet Tal. Demarcating Curves for Shape Illustration. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2008)*, 27(5) :1–9, 2008.
- [KST09] Michael Kolomenkin, Ilan Shimshoni, and Ayellet Tal. On edge detection on surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2767–2774, 2009.
- [KVHS00] Jan Kautz, Pere-Pau Vázquez, Wolfgang Heidrich, and Hans-Peter Seidel. Unified Approach to Prefiltered Environment Maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 185–196. Springer-Verlag, 2000.
- [KvK92] J.J. Koenderink, A.J. van Doorn, and A.M.L. Kappers. Surface perception in pictures. *Perception and Psychophysics*, 52 :487–496, 1992.

- [KWTM03] G Kindlmann, R Whitaker, T Tasdizen, and T Möller. Curvature-Based Transfer Functions for Direct Volume Rendering : Methods and Applications. In *Proc. IEEE Visualization 2003*, pages 513–520, October 2003.
- [KYYL08] Yongjin Kim, Jingyi Yu, Xuan Yu, and Seungyong Lee. Line-art illustration of dynamic and specular surfaces. In *SIGGRAPH Asia '08 : ACM SIGGRAPH Asia 2008 papers*, pages 1–10, New York, NY, USA, 2008. ACM.
- [LACS08] Wilmot Li, Maneesh Agrawala, Brian Curless, and David Salesin. Automated generation of interactive 3d exploded view diagrams. In *SIGGRAPH '08 : ACM SIGGRAPH 2008 papers*, pages 1–7, New York, NY, USA, 2008. ACM.
- [LAS04] Wilmot Li, Maneesh Agrawala, and David Salesin. Interactive image-based exploded view diagrams. In *GI '04 : Proceedings of Graphics Interface 2004*, pages 203–212, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.
- [LBAD⁺06] Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Inverse shade trees for non-parametric material representation and editing. *ACM Trans. Graph.*, 25(3) :735–745, 2006.
- [LCD06] Thomas Luft, Carsten Colditz, and Oliver Deussen. Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics*, 25(3) :1206–1213, jul 2006.
- [Lit97] Peter Litwinowicz. Processing images and video for an impressionist effect. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 407–414, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [LKL06] Hyunjun Lee, Sungtae Kwon, and Seungyong Lee. Real-time pencil rendering. In *NPAR '06 : Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 37–45, New York, NY, USA, 2006. ACM.
- [LM01] Eric B. Lum and Kwan-Liu Ma. Non-photorealistic rendering using watercolor inspired textures and illumination. In *PG '01 : Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, page 322, Washington, DC, USA, 2001. IEEE Computer Society.
- [LMHB00] Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. Stylized rendering techniques for scalable real-time 3d animation. In *NPAR '00 : Proceedings of the 1st international symposium on Non-*

- photorealistic animation and rendering*, pages 13–20, New York, NY, USA, 2000. ACM.
- [LMLH07] Yunjin Lee, Lee Markosian, Seungyong Lee, and John F. Hughes. Line drawings via abstracted shading. *ACM Trans. Graph.*, 26(3) :18, 2007.
- [LOG06] Sangwon Lee, Sven C. Olsen, and Bruce Gooch. Interactive 3d fluid jet painting. In *SIGGRAPH '06 : ACM SIGGRAPH 2006 Emerging technologies*, page 17, New York, NY, USA, 2006. ACM.
- [LRA⁺07] Wilmot Li, Lincoln Ritter, Maneesh Agrawala, Brian Curless, and David Salesin. Interactive cutaway illustrations of complex 3d models. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 31, New York, NY, USA, 2007. ACM.
- [LSF10] Jingwan Lu, Pedro V. Sander, and Adam Finkelstein. Interactive painterly stylization of images, videos and 3D animations. In *Proceedings of I3D 2010*, February 2010.
- [LZ09] Bruno Lévy and Hao (Richard) Zhang. Spectral mesh processing. In *SIGGRAPH ASIA '09 : ACM SIGGRAPH ASIA 2009 Courses*, pages 1–47, New York, NY, USA, 2009. ACM.
- [LZL⁺10] Liang Lin, Kun Zeng, Han Lv, Yizhou Wang, Yingqing Xu, and Song-Chun Zhu. Painterly animation using video semantics and feature correspondence. In *NPAR '10 : Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, New York, NY, USA, 2010. ACM.
- [McC94] Scott McCloud. *Understanding Comics : The Invisible Art*. Harper Paperbacks, 1994.
- [MDSB02] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds, 2002.
- [MDWK08] Sebastian Möser, Patrick Degener, Roland Wahl, and Reinhard Klein. Context aware terrain visualization for wayfinding and navigation. *Computer Graphics Forum*, 27(7) :1853–1860, oct 2008.
- [Mei96] Barbara J. Meier. Painterly rendering for animation. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 477–484, New York, NY, USA, 1996. ACM.
- [MF06] Morgan McGuire and Andi Fein. Real-time cartoon rendering of smoke and clouds. In *NPAR*, June 2006.
- [MFE07] Jason Mitchell, Moby Francke, and Dhabih Eng. Illustrative rendering in team fortress 2. In *NPAR '07 : Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 71–76, New York, NY, USA, 2007. ACM.

- [MGW01] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial texture maps. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 519–528, New York, NY, USA, 2001. ACM.
- [Mil94] Gavin Miller. Efficient Algorithms for Local and Global Accessibility Shading . In *Proc. ACM SIGGRAPH '94*, pages 319–326. ACM, 1994.
- [MKG⁺97] Lee Markosian, Michael A. Kowalski, Daniel Goldstein, Samuel J. Trychin, John F. Hughes, and Lubomir D. Bourdev. Real-time non-photorealistic rendering. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 415–420, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [MMK⁺00] Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Loring S. Holden, J. D. Northrup, and John F. Hughes. Art-based rendering with continuous levels of detail. In *NPAR '00 : Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 59–66, New York, NY, USA, 2000. ACM.
- [MYY⁺10] Niloy J. Mitra, Yong-Liang Yang, Dong-Ming Yan, Wilmot Li, and Maneesh Agrawala. Illustrating how mechanical assemblies work. *ACM Transactions on Graphics*, 29(3) :to appear, 2010.
- [ND04] Marc Nienhaus and Jürgen Döllner. Blueprints : illustrating architecture and technical parts using hardware-accelerated non-photorealistic rendering. In *Graphics Interface (GI'04)*, pages 49–56. Canadian Human-Computer Communications Society, 2004.
- [Nic70] F. E. Nicodemus. Reflectance nomenclature and directional reflectance and emissivity. *Appl. Opt.*, 9 :1474–1475, jun 1970.
- [NJLM06] Alex Ni, Kyuman Jeong, Seungyong Lee, and Lee Markosian. Multi-scale line drawings from 3d meshes. In *I3D '06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 133–137, New York, NY, USA, 2006. ACM.
- [NM00] J. D. Northrup and Lee Markosian. Artistic silhouettes : a hybrid approach. In *NPAR '00 : Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 31–37, New York, NY, USA, 2000. ACM.
- [OB10] Marc Olano and Dan Baker. Lean mapping. In *I3D '10 : Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 181–188, New York, NY, USA, 2010. ACM.
- [OBA08] James P. O'Shea, Martin S. Banks, and Maneesh Agrawala. The assumed light direction for perceiving shape from shading. In *APGV '08 : Proc. symposium on Applied perception in graphics and visualization*, pages 135–142. ACM, 2008.

- [OBBT07] Alexandrina Orzan, Adrien Bousseau, Pascal Barla, and Joëlle Thollot. Structure-preserving manipulation of photographs. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, aug 2007.
- [OBS04] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph. (Proc. SIGGRAPH 2004)*, 3(23) :609–612, 2004.
- [OJIN08] Naoto Okaichi, Henry Johan, Takashi Imagire, and Tomoyuki Nishita. A virtual painting knife. *Vis. Comput.*, 24(7) :753–763, 2008.
- [OK08] Jeff Orchard and Craig S. Kaplan. Cut-out image mosaics. In *NPAR '08 : Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 79–87, New York, NY, USA, 2008. ACM.
- [Oli05] Thomas Luft Oliver. Interactive watercolor animations, 2005.
- [Ost99] Victor Ostromoukhov. Digital facial engraving. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 417–424, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [PFH00] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [PG04] Matt Pharr and Simon Green. *GPU Gems*, chapter Ambient Occlusion. Addison-Wesley, 2004.
- [PGSP08] Romain Pacanowski, Xavier Granier, Christophe Schlick, and Pierre Poulin. Sketch and paint-based interface for highlight modeling. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*. Eurographics, jun 2008.
- [PHWF01] Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, page 581, New York, NY, USA, 2001. ACM.
- [PKS⁺01] D. L. Page, A. Koschan, Y. Sun, J. Paik, and M. A. Abidi. Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting. In *in Proceedings of VECTOR VOTING 229 the International Conference on Computer Vision and Pattern Recognition*, pages 162–167, 2001.
- [PM90] P. Perona and J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7) :629–639, July 1990.

- [PQW⁺08] Wai-Man Pang, Yingge Qu, Tien-Tsin Wong, Daniel Cohen-Or, and Pheng-Ann Heng. Structure-aware halftoning. *ACM Trans. Graph.*, 27(3) :1–8, 2008.
- [PR10] Tania Pouli and Erik Reinhard. Progressive histogram reshaping for creative color transfer and tone reproduction. In *NPAR '10 : Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pages 81–90, New York, NY, USA, 2010. ACM.
- [PS03] Koenderink J.J. Pont S.C. *Computer Analysis of Images and Patterns*, chapter Illuminance flow, pages 90–97. Springer, Berlin, 2003.
- [Ras01] Ramesh Raskar. Hardware support for non-photorealistic rendering. In *HWWS '01 : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 41–47, New York, NY, USA, 2001. ACM.
- [RBD06] Szymon Rusinkiewicz, Michael Burns, and Doug DeCarlo. Exaggerated Shading for Depicting Shape and Detail. *ACM Trans. Graph. (Proc. SIGGRAPH 2006)*, 25(3) :1199–1205, 2006.
- [RC99] Ramesh Raskar and Michael Cohen. Image precision silhouette edges. In *I3D '99 : Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 135–140, New York, NY, USA, 1999. ACM.
- [RDF05] Szymon Rusinkiewicz, Doug DeCarlo, and Adam Finkelstein. Line drawings from 3d models. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Courses*, page 1, New York, NY, USA, 2005. ACM.
- [RDF08] Szymon Rusinkiewicz, Doug DeCarlo, and Adam Finkelstein. Line drawings from 3d models. In *SIGGRAPH '08 : ACM SIGGRAPH 2005 Courses*, page 1, New York, NY, USA, 2008. ACM.
- [RMB07] Ravi Ramamoorthi, Dhruv Mahajan, and Peter Belhumeur. A First-Order Analysis of Lighting, Shading, and Shadows. *ACM Trans. Graph.*, 26(1) :2, 2007.
- [RMN05] Dave Rudolf, David Mould, and Eric Neufeld. A bidirectional deposition model of wax crayons. *Computer Graphics Forum*, 24(1) :27–39, Mar 2005.
- [RSI⁺08] Tobias Ritschel, Kaleigh Smith, Matthias Ihrke, Thorsten Grosch, Karol Myszkowski, and Hans-Peter Seidel. 3D Unsharp Masking for Scene Coherent Enhancement. *ACM Trans. Graph. (Proc. SIGGRAPH 2008)*, 27(3) :1–8, 2008.
- [Rus04] Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004.
- [RWPD05] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging : Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2005.

- [SA07] Perumaal Shanmugam and Okan Arikan. Hardware accelerated ambient occlusion techniques on gpus. In *I3D '07 : Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 73–80, New York, NY, USA, 2007. ACM.
- [SABS94] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen-and-ink illustration. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 101–108, New York, NY, USA, 1994. ACM.
- [SALS96] Mike Salisbury, Corin Anderson, Dani Lischinski, and David H. Salesin. Scale-dependent reproduction of pen-and-ink illustrations. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 461–468, New York, NY, USA, 1996. ACM.
- [SC08] Amit Shesh and Baoquan Chen. Efficient and dynamic simplification of line drawings. *Computer Graphics Forum*, 27(2), 2008.
- [Sch94a] Christophe Schlick. A Survey of Shading and Reflectance Models. *Comp. Graph. Forum*, 13(2), 1994.
- [Sch94b] Christophe Schlick. *A fast alternative to Phong's specular model*, pages 385–387. Academic Press Professional, Inc., San Diego, CA, USA, 1994.
- [Sch94c] Christophe Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum*, 13 :233–246, 1994.
- [SD08] Thierry Stein and Xavier Décoret. Dynamic label placement for improved interactive exploration. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, jun 2008.
- [Sec02] Adrian Secord. Weighted voronoi stippling. In *NPAR '02 : Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 37–43, New York, NY, USA, 2002. ACM.
- [SEH08] Matei Stroila, Elmar Eisemann, and John C. Hart. Clip art rendering of smooth isosurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 14(1) :135–145, jan 2008.
- [SGS08] Nisha Sudarsanam, Cindy Grimm, and Karan Singh. Non-linear perspective widgets for creating multiple-view images. In *NPAR '08 : Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 69–77, New York, NY, USA, 2008. ACM.
- [SKLL07] Minjung Son, Henry Kang, Yunjin Lee, and Seungyong Lee. Abstract line drawings from 2d images. In *PG '07 : Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 333–342, Washington, DC, USA, 2007. IEEE Computer Society.

- [SLK05] Kaleigh Smith, Yunjun Liu, and Allison Klein. Animosaics. In *SCA '05 : Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 201–208, New York, NY, USA, 2005. ACM.
- [SMGG01] Peter-Pike J. Sloan, William Martin, Amy Gooch, and Bruce Gooch. The lit sphere : a model for capturing npr shading from art. In *GRIN'01 : No description on Graphics interface 2001*, pages 143–150, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.
- [SMSA01] P. Shirley, S. Marschner, J. Stam, and M. Ashikhmin. State of the art in modeling and measuring of surface reflection. In *SIGGRAPH 2001 Course Notes*, 2001.
- [SSBG10] Johannes Schmid, Robert Sumner, Huw Bowles, and Markus Gross. Programmable motion effects. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 29(3), 2010.
- [ST90] Takafumi Saito and Tokiichiro Takahashi. Comprehensible Rendering of 3-D Shapes. In *Proc. ACM SIGGRAPH '90*, pages 197–206. ACM, 1990.
- [SWHS97] Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 401–406, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [SY00] Michio Shiraishi and Yasushi Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *NPAR '00 : Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 53–58, New York, NY, USA, 2000. ACM.
- [TABIO7] Hideki Todo, Ken-ichi Anjyo, William Baxter, and Takeo Igarashi. Locally controllable stylized shading. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 17, New York, NY, USA, 2007. ACM.
- [Tau95] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *ICCV '95 : Proceedings of the Fifth International Conference on Computer Vision*, page 902, Washington, DC, USA, 1995. IEEE Computer Society.
- [TFFR07] Corey Toler-Franklin, Adam Finkelstein, and Szymon Rusinkiewicz. Illustration of Complex Real-World Objects using Images with Normals. In *NPAR '07 : Proc. international symposium on Non-photorealistic animation and rendering*, pages 111–119. ACM, 2007.
- [tHR03] B.M. ter Haar Romeny. *Front-End Vision and Multi-Scale Image Analysis*. Kluwer, August 2003.

- [TKS10] Markus Tatzgern, Denis Kalkofen, and Dieter Schmalstieg. Compact explosion diagrams. In *NPAR '10 : Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pages 17–26, New York, NY, USA, 2010. ACM.
- [TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98 : Proceedings of the Sixth International Conference on Computer Vision*, page 839, Washington, DC, USA, 1998. IEEE Computer Society.
- [VBGS08a] Romain Vergne, Pascal Barla, Xavier Granier, and Christophe Schlick. Apparent relief : a shape descriptor for stylized shading. In *NPAR '08 : Proc. international symposium on Non-photorealistic animation and rendering*, pages 23–29. ACM, 2008.
- [VBGS08b] Romain Vergne, Pascal Barla, Xavier Granier, and Christophe Schlick. Shading with apparent relief. In *SIGGRAPH '08 : ACM SIGGRAPH 2008 talks*. ACM, ACM, aug 2008.
- [VBT⁺07] Romain Vergne, Adrien Bousseau, Joëlle Thollot, David Vanderhaeghe, Pascal Barla, and Xavier Granier. Utilisation du rendu expressif pour l'illustration et l'exploration de données archéologiques. In *Virtual Retrospect 2007 : Archéologie et Réalité Virtuelle*. Ausonius, 2007.
- [VBTS07] David Vanderhaeghe, Pascal Barla, Joëlle Thollot, and François Sillion. Dynamic point distribution for stroke-based rendering. In *Rendering Techniques 2007 (Proceedings of the Eurographics Symposium on Rendering)*, pages 139–146, 2007.
- [VG07] Romain Vergne and Xavier Granier. Stylisation d'objets éclairés par des cartes d'environnement hdr. In *Actes des 20èmes Journées de l'Association Française d'Informatique Graphique (AFIG)*, 2007.
- [VLD07] Peter Vangorp, Jurgen Laurijssen, and Philip Dutré. The influence of shape on the perception of material reflectance. *ACM Trans. Graph. (Proc. SIGGRAPH 2007)*, 26(3) :77, 2007.
- [VLLVR04] Tom Van Laerhoven, Jori Liesenborgs, and Frank Van Reeth. Real-time watercolor painting on a distributed paper model. In *CGI '04 : Proceedings of the Computer Graphics International*, pages 640–643, Washington, DC, USA, 2004. IEEE Computer Society.
- [Vol06] Christina Volckman. Renaissance, 2006.
- [VPB⁺09] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christophe Schlick. Light warping for enhanced surface depiction. *ACM Transaction on Graphics (Proceedings of SIGGRAPH 2009)*, Aug 2009.
- [VPB⁺10] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christophe Schlick. Radiance scaling for versatile surface en-

- hancement. In *I3D '10 : Proc. symposium on Interactive 3D graphics and games*. ACM, 2010.
- [VPB⁺11] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christophe Schlick. Improving shape depiction under arbitrary rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2011.
- [VVBB10] David Vanderhaeghe, Romain Vergne, Pascal Barla, and William Baxter. Bidirectional Appearance Distribution Function for Stylized Shading. Research Report RR-7195, INRIA, 02 2010.
- [VVJ⁺10] Romain Vergne, David Vanderhaeghe, Chen Jiazhou, Pascal Barla, Xavier Granier, and Christophe Schlick. Implicit brushes for stylized line-based rendering. Technical report, CCSD/HAL : e-articles server (based on gBUS) [<http://hal.ccsd.cnrs.fr/oai/oai.php>] (France), 2010.
- [WHA⁺07] Tim Weyrich, Simon Heinzle, Timo Aila, Daniel B. Fasnacht, Stephan Oetiker, Mario Botsch, Cyril Flaig, Simon Mall, Kaspar Rohrer, Norbert Felber, Hubert Kaeslin, and Markus Gross. A hardware architecture for surface splatting. *ACM Trans. Graph.*, 26(3) :90, 2007.
- [Wil97] John Willats. *Art and Representation : New Principles in the Analysis of Pictures*. Princeton University Press, 1997.
- [WOG06] Holger Winnemöller, Sven C. Olsen, and Bruce Gooch. Real-time video abstraction. *ACM Trans. Graph.*, 25(3) :1221–1226, 2006.
- [Woo94] Phyllis Wood. *Scientific Illustration : A Guide to Biological, Zoological, and Medical Rendering Techniques, Design, Printing, and Display*. John Wiley and Sons, Inc., New York, 2nd edition, 1994.
- [WPFH02] Matthew Webb, Emil Praun, Adam Finkelstein, and Hugues Hoppe. Fine tone control in hardware hatching. In *NPAP '02 : Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 53–ff, New York, NY, USA, 2002. ACM.
- [WS94] Georges Winkenbach and David H. Salesin. Computer-generated pen-and-ink illustration. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 91–100, New York, NY, USA, 1994. ACM.
- [WS96] Georges Winkenbach and David H. Salesin. Rendering parametric surfaces in pen and ink. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 469–476, New York, NY, USA, 1996. ACM.
- [WTXC04] Jue Wang, Bo Thiesson, Yingqing Xu, and Michael Cohen. Image and video segmentation by anisotropic kernel mean shift. In *In Proc. ECCV*, pages 238–249. Springer, 2004.

- [WXSC04] Jue Wang, Yingqing Xu, Heung-Yeung Shum, and Michael F. Cohen. Video tooning. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Papers*, pages 574–583, New York, NY, USA, 2004. ACM.
- [XCDR10] Su Xue, Xuejin Chen, Julie Dorsey, and Holly Rushmeier. Printed patterns for enhanced shape perception of papercraft models. *Computer Graphics Forum*, 29(2), 2010.
- [XHT⁺07] Xuexiang Xie, Ying He, Feng Tian, Hock-Soon Seah, Xianfeng Gu, and Hong Qin. An effective illustrative visualization framework based on photic extremum lines (pels). *IEEE Transactions on Visualization and Computer Graphics*, 13(6) :1328–1335, 2007.
- [XKM07] Jie Xu, Craig S. Kaplan, and Xiaofeng Mi. Computer-generated papercutting. In *PG '07 : Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 343–350, Washington, DC, USA, 2007. IEEE Computer Society.
- [XLSN10] Ning Xie, Hamid Laga, Sugru Saito, and Masayuki Nakajima. Ir2s : interactive real photo to sumi-e. In *NPAR '10 : Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pages 63–71, New York, NY, USA, 2010. ACM.
- [XTJ⁺08] Songhua Xu, Haisheng Tan, Xiantao Jiao, Francis C.M. Lau, and Yunhe Pan. A generic pigment model for digital painting. *Computer Graphics Forum*, 26(3) :609–618, 2008.
- [XZW10] Xuemiao Xu, Linling Zhang, and Tien-Tsin Wong. Structure-based ascii art. In *SIGGRAPH '10 : ACM SIGGRAPH 2010 papers*, pages 1–10, New York, NY, USA, 2010. ACM.
- [YBS05] Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. Fast and robust detection of crest lines on meshes. In *SPM '05 : Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 227–232, New York, NY, USA, 2005. ACM.
- [YM04] Jingyi Yu and Leonard McMillan. A framework for multiperspective rendering. In *Rendering Techniques*, pages 61–68, 2004.
- [YMS08] Jingyi Yu, Leonard McMillan, and Peter Sturm. Multiperspective modeling, rendering, and imaging. In *SIGGRAPH Asia '08 : ACM SIGGRAPH ASIA 2008 courses*, pages 1–36, New York, NY, USA, 2008. ACM.
- [Zen86] Silvano Di Zenzo. A note on the gradient of a multi-image. *CVGIP*, 33(1) :116–125, 1986.
- [ZHXC09] Long Zhang, Ying He, XueXiang Xie, and Wei Chen. Laplacian Lines for Real Time Shape Illustration. In *I3D '09 : Proc. symposium on Interactive 3D graphics and games*. ACM, 2009.
- [ZISS04] Johannes Zander, Tobias Isenberg, Stefan Schlechtweg, and Thomas Strothotte. High quality hatching. *Comput. Graph. Forum*, 23(3) :421–430, 2004.

- [ZZ10] Mingtian Zhao and Song-Chun Zhu. Sisley the abstract painter. In *NPAR '10 : Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pages 99–107, New York, NY, USA, 2010. ACM.
- [ZZXZ10] Kun Zeng, Mingtian Zhao, Caiming Xiong, and Song-Chun Zhu. From image parsing to painterly rendering. *ACM Trans. Graph.*, 29(1) :2 :1–2 :11, 2010.