

2M11.2784.9

Université de Montréal

# Analyse et synthèse de textures structurales

par

Laurent Lefebvre

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Mémoire présenté à la faculté des études supérieures  
en vue de l'obtention du grade de  
Maître ès sciences (M.Sc.)  
en informatique

août 2000

© Laurent Lefebvre, 2000



QA

76

U54

2000

n.026



Université de Montréal  
Faculté des études supérieures

Ce mémoire de maîtrise intitulé

Analyse et synthèse de textures structurales

présenté par  
Laurent Lefebvre

a été évalué par un jury composé des personnes suivantes :

Président : Neil Stewart

Directeur de recherche : Pierre Poulin

Membre : Sébastien Roy

# Sommaire

Avec l'avènement des techniques de modélisation à partir de photographies, il devient plus facile d'appliquer sur des mondes virtuels des textures extraites de la réalité. De nombreux motifs répétitifs (textures structurelles) dans les constructions humaines peuvent être reproduits à l'aide de textures procédurales. Ce genre de textures offre une puissante alternative aux textures couleur traditionnelles mais demande à l'artiste de programmer l'algorithme qui donnera les effets désirés. Nous présentons un système qui extrait semi-automatiquement les paramètres de textures structurelles tout en donnant la possibilité à l'utilisateur de guider les algorithmes.

Deux grandes classes de textures structurelles sont étudiées : le carrelage rectangulaire et le bois. Les résultats démontrent l'intérêt de synthétiser de meilleures textures à l'aide de cette technique pour des applications telles que la réalité augmentée par ordinateur.

Nous présentons aussi une méthode semi-automatique d'analyse et de synthèse de cartes de relief à partir de régions spéculaires de photographies dans le but d'augmenter le réalisme des textures générées. De nombreuses méthodes déjà existantes permettent d'extraire un relief grossier à partir de surface diffuses. Notre méthode s'applique à l'extraction de détails fins sur une surface spéculaire.

## **Mots-clés :**

textures procédurales, apposition de textures, modélisation à partir de photographies, extraction de paramètres, texture de bois, carrelage rectangulaire, réalité augmentée par ordinateur, carte de relief

# Table des matières

<b>Remerciements</b>	<b>ix</b>
<b>Remarques préliminaires</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problématique . . . . .	1
1.2 Motivation . . . . .	3
1.3 Aperçu des différents chapitres . . . . .	5
<b>2 Travaux antérieurs</b>	<b>6</b>
2.1 L'apposition de textures . . . . .	6
2.1.1 Les textures basées sur une carte d'information . . . . .	6
2.1.2 Les textures procédurales . . . . .	8
2.2 Reconstruction à partir d'images . . . . .	9
2.2.1 Projection de textures . . . . .	9
2.2.2 Extraction de textures . . . . .	10
2.3 Analyse et synthèse de textures . . . . .	11
2.3.1 L'analyse et la synthèse pyramidale de textures . . . . .	12
2.3.2 L'analyse et la synthèse multirésolution de textures . . . . .	15
2.3.3 La synthèse de texture par échantillonnage non-paramétré . . . . .	16
<b>3 Synthèse de textures structurales</b>	<b>19</b>
3.1 Le carrelage rectangulaire . . . . .	21
3.1.1 Présentation du modèle . . . . .	21
3.1.2 Extraction des paramètres du modèle . . . . .	22

3.1.3	Remplissage de l'intérieur des tuiles . . . . .	29
3.1.4	Quelques résultats du carrelage rectangulaire . . . . .	30
3.2	Le bois . . . . .	32
3.2.1	Présentation du modèle . . . . .	33
3.2.2	Extraction des paramètres du modèle . . . . .	35
3.2.3	Quelques résultats du bois . . . . .	39
<b>4</b>	<b>Synthèse de cartes de relief</b>	<b>43</b>
4.1	Notions préalables . . . . .	44
4.1.1	La BRDF . . . . .	44
4.1.2	Le modèle de Phong . . . . .	45
4.1.3	La carte de relief . . . . .	46
4.2	Travaux antérieurs . . . . .	46
4.3	Analyse et synthèse de cartes de relief . . . . .	48
4.3.1	Extraction de la carte de relief . . . . .	49
4.3.2	Le montage . . . . .	50
4.3.3	Approximation par le modèle de Phong . . . . .	50
4.3.4	Synthèse de la carte de relief . . . . .	54
4.3.5	Rendu et résultats . . . . .	54
<b>5</b>	<b>Résultats et améliorations des modèles</b>	<b>57</b>
5.1	Application de cartes de micro-déplacement et de relief . . . . .	58
5.2	Alignement entre les textures de faces adjacentes . . . . .	58
5.3	Textures procédurales et textures couleur . . . . .	60
5.4	Textures structurelles multiples . . . . .	60
5.5	Quelques exemples . . . . .	62
5.5.1	La salle de bain . . . . .	62
5.5.2	Le foyer . . . . .	65
<b>6</b>	<b>Conclusion</b>	<b>67</b>
6.1	Améliorations et extensions . . . . .	68
6.1.1	Textures structurelles hiérarchiques . . . . .	68
6.1.2	Extraction des zones d'ombre . . . . .	68

<i>TABLE DES MATIÈRES</i>	iii
6.1.3 Extraction de cartes de relief sur des objets texturés . . . . .	68
<b>A Glossaire</b>	<b>70</b>
<b>Bibliographie</b>	<b>72</b>

# Liste des tableaux

3.1 Extraction des valeurs pour le bois simulé . . . . . 41



# Liste des Algorithmes

1	Sélection d'un échantillon de tuile . . . . .	30
2	Processus d'approximation . . . . .	53

# Table des figures

1.1	Une photographie d'un foyer (à gauche) et une vue synthétique de ce même foyer (à droite) . . . . .	4
2.1	Une pyramide navigable. Image originale à gauche et pyramide navigable à droite . . . . .	13
2.2	Synthèse à l'aide de l'algorithme de Heeger et Bergen. Images d'entrée à gauche, résultat à droite . . . . .	14
2.3	Les résultats obtenus avec l'algorithme de DeBonet : (a) image originale; (b) synthèse de Heeger et Bergen [HB95]; (c) synthèse de DeBonet [De 97]	16
2.4	Les informations nécessaires à l'algorithme de Efros et Leung [EL99] : (a) image initiale; (b) filtre d'importance; (c) masque binaire; (d) semence; (e) résultat . . . . .	17
2.5	L'algorithme de Efros et Leung [EL99] : Le voisinage du pixel à synthétiser est examiné; ensuite on trouve dans l'image initiale des rectangles semblables au voisinage du pixel à synthétiser; enfin on choisit un de ces rectangles au hasard et on assigne la couleur au centre de ce rectangle au pixel à synthétiser . . . . .	17
3.1	Notre modèle de carrelage rectangulaire . . . . .	22
3.2	Texture couleur, texture segmentée (masque binaire) et image négative et agrandie de la FFT du masque notée $F(\hat{x}, \hat{y})$ . . . . .	24
3.3	Texture couleur, texture segmentée (masque binaire), négatif de la FFT du masque et agrandissement du négatif de la FFT . . . . .	25
3.4	Une photo de la tour de l'université de Montréal et une vue rapprochée qui montre les détails de la texture . . . . .	28

3.5	Synthèse d'un mur de briques : (a) image originale; (b) synthèse de DeBonet [De 97]; (c) synthèse de Efros et Leung [EL99], implantation de Wei [Wei00]; (d) synthèse de notre algorithme . . . . .	31
3.6	Synthèse d'un plancher de céramique . . . . .	32
3.7	Notre modèle de bois . . . . .	34
3.8	Texture de bois et segmentation de cette texture . . . . .	35
3.9	Différence visuelle entre plusieurs valeurs d'angle d'inclinaison : (a) $0^\circ$ ; (b) $3^\circ$ ; (c) $5^\circ$ . . . . .	37
3.10	Effet de la turbulence : (a) turbulence nulle; (b) turbulence faible; (c) turbulence moyenne . . . . .	38
3.11	Recherche de la coordonnée $X$ du centre du tronc . . . . .	39
3.12	Bois réel et bois synthétique . . . . .	40
4.1	BRDF . . . . .	45
4.2	Modèle d'illumination locale de Phong . . . . .	46
4.3	Le montage : (a) vue de côté; (b) vue arrière; (c) vue présentant l'appareil photographique ainsi que le trou d'épingle; (d) vue de l'appareil photographique . . . . .	51
4.4	(a) Image d'une réflexion spéculaire; (b) vue en perspective d'élévation de la carte de radiance reconstruite; (c) modèle de Phong [Pho75] approximant la carte de radiance; (d) carte de radiance après la soustraction des valeurs dictées par le modèle de Phong [Pho75]; (e) carte de radiance après la soustraction et filtrage passe-haut . . . . .	54
4.5	(a) image d'entrée, (b) image synthétique . . . . .	55
4.6	Trois exemples de synthèse de cartes de relief. Image d'entrée à gauche, résultat à droite . . . . .	56
5.1	Textures de briques synthétiques augmentées. À gauche une carte de relief est appliquée, à droite une carte de micro-déplacement est appliquée	58
5.2	Un vrai carrelage de céramique (à gauche) et la version synthétique de ce carrelage augmentée d'une carte de relief (à droite) . . . . .	59
5.3	Murs de briques sur une tour et vue rapprochée . . . . .	59
5.4	Vrai plancher de chêne . . . . .	61

5.5	Planchers synthétiques. Une carte de relief a été appliquée afin de courber les côtés des planches dans la figure de droite . . . . .	61
5.6	Une des photographies utilisées pour reconstruire la salle de bain . . . . .	62
5.7	La salle de bain reconstruite avec les textures projetées directement sur les murs . . . . .	63
5.8	Vue rapprochée du sol (à gauche) et d'un des murs (à droite) . . . . .	63
5.9	Trois vues sur la salle de bain avec nos textures synthétiques . . . . .	64
5.10	Substitution du plancher avec des textures extraites d'autres photographies	64
5.11	Une photographie d'un foyer . . . . .	65
5.12	Deux vues synthétiques du foyer . . . . .	66

# Remerciements

Je tiens tout d'abord à remercier mes parents sans qui je ne serais pas en train d'écrire ce texte en ce moment. Ensuite vient Evelyne qui a su me supporter pendant ces deux années. Que dire de Pierre Poulin ? Ma foi, que c'est un directeur très présent et à l'écoute de ses étudiants ? Qu'il fut très plaisant de travailler en sa compagnie ? Disons simplement merci, et espérons que cela suffise... Je tiens aussi à remercier tous les autres membres du laboratoire pour les discussions scientifiques et non-scientifiques que nous avons pu avoir. La DGTIC m'a aussi donné un fier coup de main en léguant blinn, notre Onyx, au laboratoire. Je me dois aussi de remercier le CRSNG pour avoir financé cette recherche. La cafetière du laboratoire a été très utile lors du premier été de la recherche mais elle a malheureusement rendu l'âme suite à un coup sournois de Nicolas. Les balles de jonglerie ont su rendre toute compilation une partie de plaisir et le cube rubik n'est finalement pas si difficile que ça !

Du côté des compagnies, Christophe Bernaud autant chez Behavior que maintenant chez lcestorm digital m'a fourni les ressources et l'équipement photographique nécessaire à certaines parties de ma recherche.

# Remarques préliminaires

Certains mots qui peuvent porter à confusion ou qui ont été empruntés de l'anglais sont définis au glossaire à la page 70. La première occurrence de ces mots est mise en évidence par une *police de caractères oblique*.

Le système d'analyse et de synthèse de *textures structurelles* a été implanté sur Irix 6.5 en utilisant comme base la réimplantation du système Rekon [POF98] développée par Martin Blais, Laurent Lefebvre et Mathieu Ouimet. Pour la présentation dans le mémoire, toutes les images ont été converties en 256 tons de gris.

À travers ce texte, toutes les expressions mathématiques en **caractères gras** sont des vecteurs.

# Chapitre 1

## Introduction

Nous décrivons brièvement dans ce chapitre, les problèmes qui ont motivé cette recherche, les buts de notre travail ainsi que les solutions que nous proposons. Nous terminons en donnant un aperçu des différents chapitres de l'ouvrage.

### 1.1 Problématique

En *informatique graphique*, afin de simuler la complexité des scènes réelles à peu de frais, les artistes utilisent très souvent des *textures*. Ces textures encodent des détails de surface et sont apposées sur un objet grâce à une paramétrisation de sa surface. Elles peuvent grossièrement être classées sous deux grandes catégories : les *textures basées sur une carte d'information* et les *textures procédurales*.

Les textures basées sur une carte d'information sont les textures les plus fréquentes en informatique graphique ; une carte d'information (discrétisation en éléments uniformes d'une information quelconque qui peut être l'orientation des normales à la surface ou la micro-géométrie d'un objet ; bien souvent cette carte d'information se résume à une image couleur) est utilisée pour modifier les données concernant un polygone ou un objet en utilisant une paramétrisation donnée sur sa surface. Un exemple courant de l'utilisation de textures basées sur une carte d'information est l'utilisation d'une image pour modifier les couleurs d'un objet. Les grands avantages des textures basées sur une carte d'information sont leur simplicité d'utilisation et le fait que cette méthode est implantée en *matériel graphique* sur bon nombre de cartes graphiques, ce qui rend

leur utilisation très rapide. Cependant, dépendant de la paramétrisation de la surface, la jonction entre deux faces d'un objet se fait parfois plus ou moins bien et la texture a souvent l'air d'être peinte sur l'objet. Pour les applications qui demandent un haut taux d'images par seconde comme les jeux, bien souvent on ne modifie que la couleur des objets (à l'aide de textures) de la scène en utilisant une image couleur comme carte d'information. Dans ce cas particulier nous parlerons de *textures couleur*, qui sont un sous-ensemble de la catégorie plus générale des textures basées sur une carte d'information.

Les textures procédurales diffèrent des textures basées sur une carte d'information car plutôt que d'utiliser une carte d'information, un algorithme modifie les valeurs des paramètres d'un objet. Cette façon de faire donne des textures généralement plus compactes et permet souvent de faire une jonction parfaite entre les faces d'un objet. Par exemple, le bois et le marbre sont souvent reproduits par une texture procédurale car cela permet de créer des objets qui semblent avoir été taillés dans un bloc de bois ou de marbre. Cependant, il incombe à l'artiste de créer l'algorithme qui donnera le résultat escompté. De plus, l'utilisation de ces textures est souvent moins rapide que les textures basées sur une carte d'information car il faut évaluer une procédure qui peut être assez complexe à chaque fois que l'on veut avoir la couleur (ou le micro-relief) de l'objet texturé en un point donné.

En pratique, les deux types de textures présentés aux paragraphes précédents sont souvent utilisés en conjonction. Par exemple, il est très fréquent d'utiliser une texture couleur pour modifier les couleurs d'un objet et d'utiliser en même temps une texture procédurale afin de modifier le relief de ce même objet. Lorsqu'un haut degré de réalisme est nécessaire, de nombreuses couches de textures de l'un ou l'autre type sont appliquées les unes par dessus les autres afin de simuler les couleurs et le micro-relief de l'objet réel que l'on essaie de reproduire. Il est aussi possible que la texture couleur serve de valeur d'entrée pour une texture procédurale. Par exemple, la couleur plus pâle du béton peut déterminer un déplacement de surface vers l'intérieur dans un mur de briques.

Avec l'avènement de la modélisation à partir de photographies, il devient très intéressant d'extraire directement les textures à partir des photographies utilisées pour la modélisation. Dans ce contexte, de nombreux problèmes peuvent se poser :



1. Des parties de la texture que l'on veut extraire peuvent être cachées par d'autres objets présents dans la photographie.
2. Il peut y avoir des différences d'illumination entre les photographies dues à l'ombre, aux reflets, ou à d'autres facteurs.
3. La résolution de la photographie peut être insuffisante.

Afin de répondre aux problèmes d'occlusion, de nombreux chercheurs [HB95, De 97, EL99, WL00] ont proposé des algorithmes d'analyse et de synthèse de textures afin de remplir les trous dans la texture causés par un manque d'information ou simplement de créer une texture ressemblant à un échantillon donné. Cependant, ces algorithmes, bien que généraux et complètement automatiques, ont tous de la difficulté à synthétiser des textures qui présentent des motifs répétitifs car une légère erreur lors de l'application de ces motifs saute tout de suite aux yeux de l'observateur. Dans le cadre de cette recherche, nous nous sommes penchés sur la détermination semi-automatique des valeurs des paramètres de textures ainsi que la synthèse de ces textures qui présentent des motifs répétitifs et que nous appelons des textures structurelles.

## 1.2 Motivation

Les textures structurelles sont très présentes dans les environnements urbains. Par exemple, les murs de briques, plafonds suspendus, mosaïques, le bois, les vitres des gratte-ciel font tous partie de cette catégorie. Le but de notre recherche est de trouver une méthode d'analyse et de synthèse de textures structurelles afin de générer des textures d'une bonne qualité sur des scènes extraites à partir de photographies.

Pour ce faire, nous proposons une méthode semi-automatique d'analyse et de synthèse de deux grandes classes de textures structurelles : le carrelage rectangulaire et le bois. Afin d'enrayer plusieurs problèmes liés à l'extraction directe de textures à partir de photographies, nous proposons d'extraire les paramètres de la texture plutôt que les couleurs pour pouvoir générer cette dernière à notre guise sans les défauts liés à l'extraction directe des couleurs de la texture.

En effet, en informatique graphique, de nombreux algorithmes ont été écrits sous forme de textures procédurales afin de simuler à la fois le bois et le carrelage rectangulaire. Un problème de ces textures procédurales est qu'il faut ajuster à la main chacun

des paramètres requis par les procédures afin d'obtenir le résultat escompté. Nous nous proposons donc de déduire des valeurs pour ces paramètres à partir de vraies photographies de bois et de carrelages rectangulaires afin d'avoir une texture qui ressemble à celle prise en photographie sans avoir à ajuster les paramètres d'une procédure qui crée ce type de texture. La méthode que nous proposons extrait donc d'une photographie les paramètres d'une texture procédurale afin de profiter de tous les avantages inhérents à l'utilisation d'une texture procédurale.

Par exemple, la Figure 1.1 illustre notre méthode. La procédure prend comme entrée une ou plusieurs photographies d'un objet réel. Premièrement, nous utilisons notre logiciel de reconstruction à partir de photographies [POF98] afin d'extraire la géométrie des objets de la scène photographiée. Ensuite, les informations concernant les textures contenues dans cette photographie sont extraites par nos algorithmes (taille des briques, alignement de ces dernières, taille des planches, couleurs du bois, etc.). Cette façon de faire nous permet de générer procéduralement des textures similaires d'une très bonne qualité comme l'illustre la partie de droite de la Figure 1.1.



FIG. 1.1: Une photographie d'un foyer (à gauche) et une vue synthétique de ce même foyer (à droite)

Notre méthode possède plusieurs avantages :

1. La texture est définie procéduralement, ce qui permet de la générer à la résolution voulue, de répéter un motif à l'infini, et d'aligner la texture correctement sur les faces adjacentes d'un objet.
2. Étant donné que nous avons une approximation des paramètres de la texture plutôt que ses couleurs, il est plus simple de générer une texture sans trou.

3. La texture procédurale prend moins de mémoire qu'une texture basée sur une carte d'information standard car il ne s'agit que de stocker quelques paramètres afin d'être en mesure de la générer.

4. Étant donné que nous avons les paramètres de la texture, il est plus facile de l'augmenter car nous savons où se situent les divers éléments qui constituent la texture (le ciment et les tuiles par exemple dans le cas du carrelage rectangulaire).

Cependant, la méthode que nous proposons a aussi quelques inconvénients tels que :

1. Notre méthode est limitée aux textures représentant un carrelage rectangulaire ou du bois.

2. La méthode d'extraction n'est pas complètement automatique, elle demande à l'utilisateur de créer un masque binaire qui sépare les différents éléments présents dans la texture.

3. Étant donné le nombre limité de paramètres extraits, les textures synthétisées sont souvent trop parfaites (surtout dans le cas du carrelage rectangulaire).

### 1.3 Aperçu des différents chapitres

Dans le Chapitre 2, nous introduirons les différents travaux préalables dans les domaines reliés à notre recherche. Le Chapitre 3 décrit en détail l'approche que nous proposons afin de faire l'analyse et la synthèse de deux grandes classes de textures structurelles : le carrelage rectangulaire et le bois. Le Chapitre 4 propose une méthode d'analyse et de synthèse de *cartes de relief* qui peut être utilisée afin d'augmenter le réalisme de nos textures. Enfin, le Chapitre 5 montre les résultats obtenus avec nos textures structurelles sur divers modèles géométriques extraits à partir de photographies.

## Chapitre 2

# Travaux antérieurs

*La grandeur des actions humaines se mesure à l'inspiration qui les fait naître.*

*Louis Pasteur*

Dans ce chapitre, nous exposons les travaux qui touchent *l'apposition de textures*, la reconstruction à partir d'images ainsi que l'extraction de textures à partir d'images. À travers notre description de ces techniques nous parlerons aussi des avantages et des inconvénients de ces dernières.

Le chapitre se divise en trois sections : l'apposition de textures, la reconstruction à partir d'images, l'analyse et la synthèse de textures. Les deux premières sections ne seront pas étudiées en profondeur, leur principale intention étant de situer le lecteur. La dernière section sera vue en plus grand détail en raison de son impact direct sur les travaux présentés dans cet ouvrage.

### 2.1 L'apposition de textures

Dans cette section, nous présentons sommairement les différentes méthodes d'apposition de textures.

#### 2.1.1 Les textures basées sur une carte d'information

L'apposition de textures couleur (qui sont un sous-ensemble des textures basées sur une carte d'information) sur une surface est une technique qui a été introduite

par Catmull [Cat74] et ensuite raffinée par Blinn et Newell [BN76]. Cette technique consiste à utiliser une image digitale et à la “coller” sur une surface afin de donner une illusion de complexité. La surface est paramétrée en deux dimensions ainsi que l’image à apposer sur cette dernière. Ensuite, pour chaque pixel à afficher, il faut savoir comment ce dernier se projette sur la surface tridimensionnelle et ultimement comment la partie de surface considérée se transforme sur l’image digitale (qui représente les couleurs de la texture). On doit ensuite analyser la partie de la texture ainsi couverte afin de savoir quelle couleur donner au pixel à afficher. Cette partie du traitement est la plus importante car si l’analyse est mal faite [Whi85] elle peut laisser place à beaucoup d’*aliassage* et une à mauvaise reconstruction de la texture. La technique d’apposition de textures est encore aujourd’hui la technique la plus utilisée à cause de sa simplicité. Pour cette raison, de nombreux travaux ont été faits sur le filtrage des textures couleur afin de réduire l’aliassage. Ainsi, plutôt que d’utiliser un simple filtre boîte, Blinn [Bli89] propose l’utilisation d’un filtre sinusoïdal qui réduit l’aliassage. Williams [Wil83] propose de pré-filtrer l’image initiale afin d’accélérer le rendu de textures couleur. Pour le lecteur intéressé, un bon survol des méthodes de filtrage a été rédigé par Heckbert [Hec86].

La technique d’apposition de textures couleur présente de nombreux avantages comme par exemple la simulation de complexité à faible coût, la facilité d’implantation ainsi que la généralité de la méthode et le fait de pouvoir filtrer la texture. Cependant de nombreux problèmes, principalement reliés au fait que l’image utilisée comme texture est de résolution finie, sont inhérents à la technique. En effet, étant donné que la résolution de la texture est finie, on ne peut s’approcher trop de l’objet texturé sans voir apparaître des artifices dus à la résolution de la texture. De plus, d’autres problèmes sont rencontrés aux jonctions entre différentes textures. Par exemple, si une unique texture de bois est posée sur chaque face d’un cube, les jonctions entre les faces du cube ne seront pas naturelles et le cube aura l’air d’avoir été peint avec la texture de bois plutôt que découpé dans une pièce de bois. Finalement, il est nécessaire de paramétrer la surface afin d’apposer correctement la texture et il faut éviter les effets de l’illumination réelle (reflets, ombre, etc.) sinon des incohérences apparaîtront au moment du rendu lorsque l’illumination synthétique se mélangera avec l’illumination présente

dans la texture couleur (ce point est surtout important lorsque l'on prend les textures directement à partir de photographies de modèles réels comme dans la Section 2.2.2).

Afin d'augmenter le réalisme des textures, en plus de simplement altérer les couleurs on peut modifier d'autres paramètres de la surface. Blinn [Bli78] a introduit une technique de perturbation de normales à la surface appelée la carte de relief afin de simuler des effets de modifications géométriques (changements de réflexion) sans toutefois modifier la géométrie de l'objet. Cette technique donne des effets convaincants mais elle ne modifie pas la silhouette de l'objet. De plus, il est difficile de filtrer les données contenues dans la carte de relief (voir les travaux de Fournier [Fou92] et Max [BM93] à ce sujet) afin de diminuer les effets de l'aliassage étant donné que les valeurs contenues dans la carte de relief ne correspondent pas linéairement aux intensités des pixels (le filtrage de la carte de relief ne fait que lisser la surface). Cook [Coo84] introduisit par la suite la *carte de micro-déplacement* dans laquelle la surface est vraiment déplacée le long des normales de la surface afin de régler les problèmes de modification de la silhouette et d'auto-occultation. Notons cependant que cette technique ne règle pas les problèmes de l'aliassage car le filtrage est tout aussi problématique en 3D qu'il l'était dans le cas des cartes de relief. De plus, la synthèse d'image par lancer de rayons est difficile lorsque l'on se sert de cartes de micro-déplacement à cause du coût de calcul important dû au fait qu'il faut intersecter un grand nombre de micro-polygones pour chaque surface.

### 2.1.2 Les textures procédurales

Afin de régler le problème des jonctions entre deux ou plusieurs faces, Peachey [Pea85], Perlin [Per85] et Perlin et Hoffert [PH89] ont introduit le concept de textures procédurales tridimensionnelles. Les textures procédurales, contrairement aux textures apposées, sont calculées à l'aide d'un algorithme et d'une paramétrisation 3D simple. Elles ont l'avantage de ne pas souffrir des artifices dus à la résolution mais ne peuvent cependant pas reproduire tous les effets que les textures couleur véhiculent. De plus, il incombe à l'artiste de rédiger l'algorithme qui reproduit les effets voulus. Depuis ces travaux [Per85, Pea85, PH89], les textures procédurales ont été raffinées par de nombreux auteurs. Par exemple, Fleisher *et al.* [FLCB95] proposent un algorithme

procédural de génération de textures cellulaires afin de simuler les textures d'apparence biologique (peau, écailles, poil, etc.). Pour le lecteur intéressé, le livre de Ebert *et al.* [EMP<sup>+</sup>94] décrit un bon nombre de textures procédurales et leurs applications.

## 2.2 Reconstruction à partir d'images

Avec l'avènement des photographies digitales, il devient de plus en plus attrayant de modéliser des scènes à partir de photographies. Par exemple, à partir de multiples photographies d'un objet, on veut pouvoir déduire un modèle géométrique tridimensionnel identique pour pouvoir par la suite le modifier ou le regarder à des angles différents de ceux des photographies. Cette façon de faire permet d'obtenir des textures et des modèles géométriques plus étroitement liés à de vrais objets tout en minimisant le travail requis pour créer à la fois le modèle géométrique et les textures qui seront appliquées sur ce modèle.

Un problème se pose cependant : les textures présentes dans les photographies sont déformées à cause de la projection en perspective causée par l'appareil photographique et de multiples photographies sont souvent nécessaires à la reconstruction d'une texture car une seule photographie ne contient que l'information sur les surfaces visibles à partir du point de vue de la photographie.

Dans ce contexte, trois options s'offrent à l'utilisateur pour le traitement des textures contenues dans les photographies : les projeter sur le modèle tridimensionnel en fonction d'un nouveau point de vue, les extraire de toutes les photographies et essayer de les combiner afin de ne former qu'une seule texture ou encore déduire les paramètres de la texture présente sur les photographies pour ensuite pouvoir générer cette texture procéduralement. Les deux premières méthodes sont présentées dans les deux sections suivantes et produisent des textures basées sur des cartes d'information. La dernière méthode est celle que nous proposons et génère une texture procédurale. Cette méthode sera présentée au Chapitre 3.

### 2.2.1 Projection de textures

Cette technique de rendu des textures est dépendante du point de vue. En fonction du point de vue, les textures des photographies sont projetées et interpolées sur

le modèle afin de produire l'aspect le plus réaliste possible lors de la synthèse d'une scène tridimensionnelle. La méthode de projection de textures est analogue à un projecteur qui éclaire le modèle géométrique à travers les photographies. Le système de modélisation à partir d'images *Façade* introduit par Debevec *et al.* [DTM96] utilise cette technique afin d'effectuer le rendu des textures contenues dans les photographies. La technique a l'avantage de produire des textures très réalistes à faible coût, de ne demander aucune paramétrisation et d'avoir une bonne continuité entre les surfaces adjacentes du modèle synthétique. Cependant, le faible nombre de photographies utilisées pour modéliser la scène peut causer des "trous" dans les textures et il est aussi possible que, plusieurs photographies se chevauchant, des artifices (fantômes sur une même surface dus aux chevauchements des textures sur la surface, même texture sur plusieurs surfaces avoisinantes, etc.) apparaissent dans les textures projetées (ces problèmes sont typiques aux méthodes présentées aux Sections 2.2.1 et 2.2.2 et sont très difficiles à résoudre). De plus, la projection de la texture est fonction de l'orientation du point de vue et de l'aire de la surface, ce qui complique les calculs (d'autant plus qu'il faut projeter le même nombre de textures qu'il y a de photographies afin de combler les "trous" possibles et enrayer les différents artifices présents dans les photographies). Debevec *et al.* [DYB98] ont amélioré la technique de projection de textures en précalculant quelques textures par polygone ce qui leur permet d'utiliser le matériel graphique existant pour afficher leurs textures projetées et ainsi afficher les textures beaucoup plus rapidement.

Havaldar *et al.* [HLM96] utilisent une technique similaire à celle de Debevec *et al.* [DTM96] mais cette fois on projette les primitives dans les images avec une transformation projective d'image à image plutôt que de monde à image et on ne considère qu'une seule photographie par texture projetée.

### 2.2.2 Extraction de textures

L'autre façon de faire le rendu de textures contenues dans des photographies digitales est d'extraire toutes les textures présentes dans les différentes photographies représentant une face donnée du modèle tridimensionnel pour ensuite les unifier en une seule texture couleur. De cette façon, on peut tirer avantage de l'implantation



matérielle des textures couleur dans les systèmes de synthèse d'image traditionnels. Cette méthode est utilisée dans le système *Rekon* de Poulin *et al.* [POF98]. L'avantage marqué de cette technique par rapport aux textures projetées est que l'on se retrouve avec une série de textures corrigées de la distorsion de perspective qui contiennent une variété d'informations engendrées par les différents points de vue. Ces différences de couleur pourront servir à une étude plus approfondie afin d'arriver à construire une texture unifiée de grande qualité. La technique pose cependant d'importants problèmes. En effet le processus d'unification de textures est loin d'être simple et l'intervention de l'utilisateur peut être nécessaire afin de mener à bien cette opération. De plus, à cause d'erreurs de modélisation ou numériques, les textures se joignent très rarement parfaitement aux arêtes des objets et certains artifices dus à l'illumination peuvent être encore présents dans les textures unifiées.

Les textures corrigées de la distorsion de perspective générées dans la première étape de cette technique serviront d'entrée à notre algorithme d'analyse et synthèse de textures structurelles présenté au Chapitre 3.

### 2.3 Analyse et synthèse de textures

Dans cette section, nous présentons en ordre chronologique les travaux antérieurs les plus récents sur l'analyse et la synthèse de textures. Les algorithmes présentés dans cette section ont tous la particularité de prendre une image d'entrée pour ensuite générer une image qui peut être plus grande et qui ressemble à l'image d'entrée. Les principales applications de ces algorithmes sont le remplissage de "trous" présents dans une texture ou simplement l'agrandissement d'une texture afin d'avoir une meilleure résolution. Les algorithmes présentés sont tous automatiques (ils ne demandent aucune intervention de la part de l'utilisateur) mais ne fonctionnent vraiment bien que sur des textures qui ne présentent pas de motifs répétitifs simples. Nous appellerons ces textures des *textures stochastiques*. Étant donné que tous ces algorithmes ne génèrent que des images et non des procédures, les textures ainsi générées tombent toutes dans la catégorie des textures basées sur une carte d'information.

Dans ce domaine, les travaux précédant ce que nous présentons dans cette section touchent principalement les modèles de réaction-diffusion [Tur91, WK91], les modèles

dans le domaine fréquentiel [Lew84], les modèles fractals [Lew87, FFC82] et les modèles de champs statistiques et aléatoires [CJ83, GM85, CK85, Gag87, BG89, JFP93, PP93]. Nous entrerons plus dans les détails dans cette section étant donné que ces travaux sont très reliés à ce que nous présentons au Chapitre 3.

### 2.3.1 L'analyse et la synthèse pyramidale de textures

Heeger et Bergen [HB95] proposent une méthode pyramidale d'analyse et de synthèse de textures stochastiques. La méthode est basée sur une représentation pyramidale de la texture proposée par Simoncelli *et al.* [SFAH92] appelée la *pyramide navigable*. Cette représentation est une transformation linéaire de l'image en une somme pondérée de fonctions de base. Formellement, si  $I(x, y)$  est une image, on peut la représenter comme une somme d'une collection donnée de fonctions indexées  $g_i(x, y)$  :

$$I(x, y) = \sum_i y_i g_i(x, y) \quad (2.1)$$

où  $y_i$  sont les coefficients de la transformation. Ces coefficients sont calculés à partir de l'image originale  $I(x, y)$  en projetant cette dernière sur un ensemble de fonctions de projection  $h_i(x, y)$  :

$$y_i = \sum_{x, y} h_i(x, y) I(x, y). \quad (2.2)$$

Contrairement à la plupart des transformées en ondelettes discrètes, la pyramide navigable n'est pas formée de fonctions de base orthogonales et est redondante : le nombre de pixels utilisés pour représenter la pyramide dépasse largement le nombre de pixels de l'image originale. Ceci sert à réduire l'aliasage à travers chaque sous-bande. Cette propriété est très importante car l'algorithme de Heeger et Bergen [HB95] traite chaque sous-bande de la pyramide indépendamment.

En pratique, une pyramide navigable est construite en filtrant l'image d'entrée à l'aide de filtres passe-bas ou passe-bandes afin d'obtenir les images qui constituent les différents niveaux de la pyramide. De plus, la pyramide navigable divise chacune de ses bandes de fréquences en une série de bandes d'orientation, ce qui permet la synthèse

de textures anisotropes (textures qui présentent une orientation préférée comme des cheveux peignés par exemple). La Figure 2.1 donne un exemple visuel d'une pyramide navigable de trois niveaux, les chiffres indiquent le niveau de la pyramide (qui correspondent aux indices des fonctions indexées  $g_i(x, y)$ ), 1 étant la racine de la pyramide. Notez que la portion blanche en haut à gauche de la pyramide ne fait pas partie de cette dernière, il ne s'agit que d'un remplissage pour rendre l'image carrée. Sur la Figure 2.1, les bandes de fréquences sont représentées par des chiffres différents alors que les bandes d'orientation sont représentées par des chiffres identiques. Par exemple, au niveau 2 de la pyramide (2 étant la deuxième bande de fréquence) nous avons 3 bandes d'orientation (il y a trois chiffres 2 sur l'image).

Des expériences psychophysiques et physiologiques suggèrent que les informations contenues dans une image sont représentées dans le cortex visuel par des orientations et des filtres sélectifs spatiaux et fréquentiels que la pyramide navigable capture en partie. Ainsi, les images synthétisées à l'aide de la pyramide navigable seront meilleures que celles synthétisées à l'aide d'une autre transformation sans ces propriétés (comme la pyramide Laplacienne par exemple [BA83]).

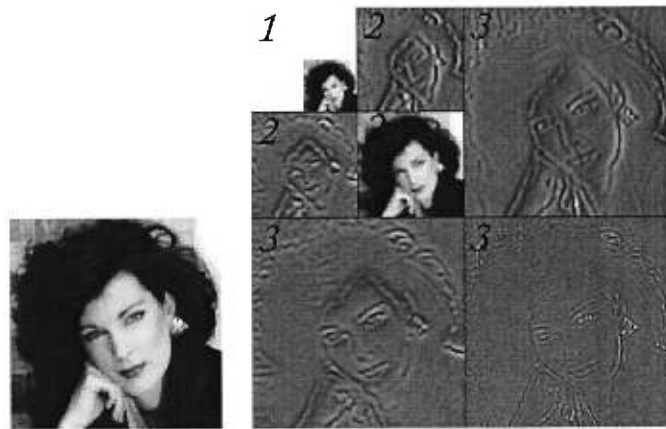


FIG. 2.1: Une pyramide navigable. Image originale à gauche et pyramide navigable à droite

L'algorithme d'analyse et de synthèse proposé par Heeger et Bergen [HB95] consiste à faire de la mise en correspondance entre l'histogramme de l'image originale et l'histogramme d'une image de bruit blanc. La mise en correspondance des histogrammes est une généralisation de l'égalisation d'histogramme (procédé utilisé en traitement

d'images pour mieux répartir les couleurs d'une image afin de faire ressortir les contrastes présents dans cette image). L'algorithme prend une image de bruit blanc et la contraint, à l'aide d'une paire de tables de correspondances, à avoir un histogramme particulier (semblable à l'image d'entrée que l'on cherche à reproduire). La mise en correspondance des histogrammes se fait à travers toutes les bandes et sous-bandes d'orientations et de fréquences des pyramides navigables de l'image de bruit blanc et de l'image d'entrée et ce, de multiples fois (cinq répétitions semblent donner de bons résultats aux dires des auteurs). Il est possible à l'aide de cet algorithme de générer à la fois des textures stochastiques bidimensionnelles et tridimensionnelles (par tranches). De plus, on peut créer un mélange entre deux textures en prenant la pyramide navigable de deux textures différentes plutôt que de ne prendre qu'une texture et une image de bruit blanc. La Figure 2.2 présente un exemple de synthèse à l'aide de l'algorithme de Heeger et Bergen [HB95].

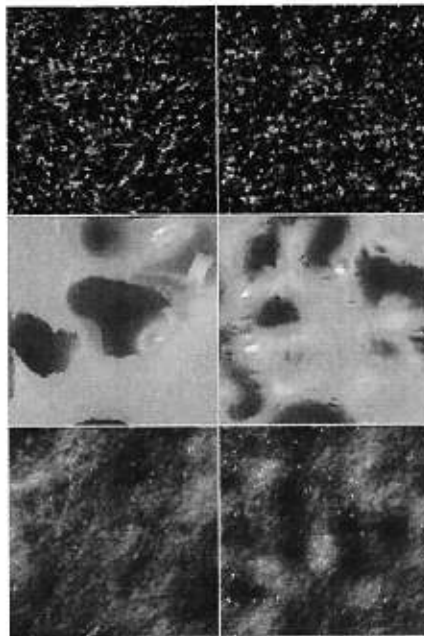


FIG. 2.2: Synthèse à l'aide de l'algorithme de Heeger et Bergen. Images d'entrée à gauche, résultat à droite

### 2.3.2 L'analyse et la synthèse multirésolution de textures

DeBonet [De 97] propose une méthode d'analyse et de synthèse multirésolution de textures basée sur une pyramide Laplacienne [BA83]. Cette pyramide ressemble beaucoup à la pyramide navigable mais utilise des filtres approximativement radialement symétriques.

Plutôt que de simplement mettre en correspondance les histogrammes de chaque niveau de la pyramide comme dans le cas de Heeger et Bergen [HB95], DeBonet [De 97] utilise un algorithme d'échantillonnage plus complexe en vue de minimiser la différence perceptuelle entre les caractéristiques des images tout en maximisant la différence perceptuelle entre l'image d'entrée et l'image synthétique. Cet algorithme assure une image qui a les mêmes caractéristiques que celle en entrée sans toutefois être identique. Concrètement, cela veut dire que l'on aura grâce à cet algorithme, une image qui ressemble à l'image d'entrée sans être pareille (un peu comme si l'on prend deux bouts de tissu provenant de la même grande pièce de tissu, il vont se ressembler mais ne seront pas identiques).

L'échantillonnage de l'image d'entrée est basé sur l'hypothèse qu'une image est perçue semblable à une autre si, sous une certaine résolution, elles contiennent des régions qui diffèrent moins qu'un certain seuil donné. Si le seuil est bien choisi (le seuil définit jusqu'à quel point l'image d'entrée est perturbée, on veut un seuil strict pour des images semi-structurelles et assez élevé pour des images stochastiques), mélanger ces régions aléatoirement ne changera pas les caractéristiques perçues de la texture tout en maximisant la différence visuelle entre l'image d'entrée et l'image synthétique. Si on applique ce concept sur la représentation pyramidale Laplacienne d'une image [BA83] (version simplifiée de la pyramide navigable présentée à la Section 2.3.1), cela correspond à modifier la position des bandes de basses fréquences aléatoirement afin d'altérer l'apparence visuelle de l'image, mais aussi à modifier en conséquence les bandes de hautes fréquences de la pyramide afin de ne pas altérer les caractéristiques de l'image.

Étant donné que seules les contraintes locales (le fait de mélanger les niveaux supérieurs de la pyramide Laplacienne correspond à ne prendre en compte que les contraintes de correspondance locales de l'image) sont utilisées par ce type

d'échantillonnage, toute image avec une structure complexe non-stochastique (comme par exemple une texture structurée qui a une structure bien précise qui se répète à travers l'image) ne sera pas synthétisée correctement par cet algorithme. Malgré tout, cet algorithme donne de meilleurs résultats que celui de Heeger et Bergen [HB95] dans presque tous les cas. La Figure 2.3 montre les résultats obtenus avec l'algorithme de DeBonet [De 97] en comparaison avec ceux obtenus avec l'algorithme de Heeger et Bergen [HB95] (les deux images (b) et (c) sont générées complètement synthétiquement à partir de l'échantillon (a)).

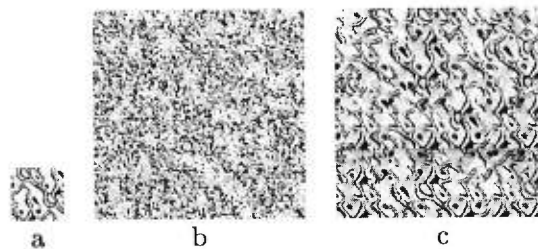


FIG. 2.3: Les résultats obtenus avec l'algorithme de DeBonet : (a) image originale; (b) synthèse de Heeger et Bergen [HB95]; (c) synthèse de DeBonet [De 97]

### 2.3.3 La synthèse de texture par échantillonnage non-paramétré

Efros et Leung [EL99] proposent un algorithme de synthèse basé sur un échantillonnage non-paramétré. À la différence des algorithmes décrits jusqu'ici, cette méthode fait grandir une image à partir d'une semence, un pixel à la fois. L'hypothèse de base est que l'image originale suit un champ aléatoire de Markov et donc, la couleur d'un pixel donné dépend de ses voisins (le filtre d'importance spatiale est défini par l'utilisateur). De plus, contrairement aux techniques présentées aux deux sections précédentes, l'image d'entrée sert de semence à la création de la texture synthétique, elle sera donc contenue dans la texture synthétique.

L'idée générale de la procédure est facile à comprendre. L'utilisateur crée un masque indiquant quelle section de l'image est la semence et quelle partie de l'image est à synthétiser, ainsi qu'un filtre qui définit l'importance relative des voisins sur le pixel à synthétiser. La Figure 2.4 montre les informations nécessaires à l'algorithme de Efros et Leung [EL99].

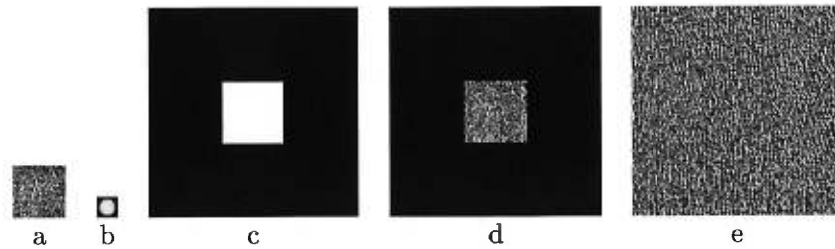


FIG. 2.4: Les informations nécessaires à l'algorithme de Efros et Leung [EL99] : (a) image initiale; (b) filtre d'importance; (c) masque binaire; (d) semence; (e) résultat

Ensuite, étant donné la texture initiale et un pixel à synthétiser qui se situe en périphérie de la semence, l'algorithme recherche tous les rectangles similaires à la zone entourant le pixel à synthétiser de la taille du filtre d'importance pondérés par ce dernier dans la texture initiale. Pour finir, un rectangle est choisi au hasard parmi ceux trouvés dans l'image initiale et la couleur du pixel au centre de ce dernier est choisie comme la couleur du pixel à synthétiser. La Figure 2.5 illustre cette démarche. Le processus se répète tant qu'il reste des pixels à synthétiser dans l'image.

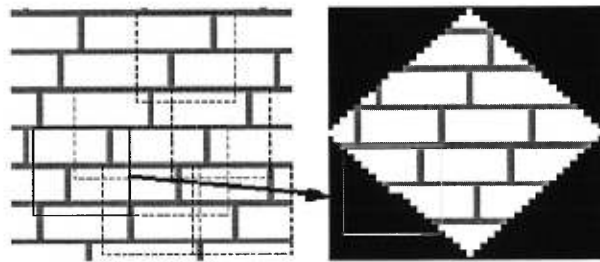


FIG. 2.5: L'algorithme de Efros et Leung [EL99] : Le voisinage du pixel à synthétiser est examiné; ensuite on trouve dans l'image initiale des rectangles semblables au voisinage du pixel à synthétiser; enfin on choisit un de ces rectangles au hasard et on assigne la couleur au centre de ce rectangle au pixel à synthétiser

Cet algorithme donne de très bons résultats visuels (comme l'atteste la Figure 2.5) mais encore une fois ne réussit pas à synthétiser les textures structurales correctement comme nous le verrons ultérieurement à l'aide d'un exemple. De plus, comme on peut s'en douter, le processus de synthèse peut être très long à cause de la recherche à faire pour chacun des pixels à synthétiser.

Plus récemment, Wei et Levoy [WL00] ont proposé une technique similaire à Efros

et Leung [EL99] mais qui gagne jusqu'à deux ordres de grandeur en vitesse grâce à une quantification sous forme d'arbre structuré vectoriel. Par exemple, sur un R10000 à 195 Mhz, pour une image initiale de  $64 \times 64$  et une image synthétique à générer de  $192 \times 192$  avec un filtre d'importance de  $9 \times 9$ , l'algorithme standard de Efros et Leung [EL99] prend 1941 secondes pour faire la synthèse alors que l'algorithme de Wei et Levoy [WL00] ne prend que 503 secondes en recherche exhaustive et 12 secondes en recherche sommaire (en échantillonnant un arbre vectoriel et en dégradant légèrement la qualité de la texture synthétique).

Dans ce chapitre, nous avons présenté les travaux antérieurs portant sur les méthodes d'apposition de textures ainsi que ceux touchant l'analyse et la synthèse de textures. Comme nous avons pu le voir, aucune des méthodes présentées n'est parfaite. En effet, bien que les méthodes d'analyse et de synthèse présentées dans ce chapitre soient générales, elles sont toutes incapables de synthétiser correctement des textures structurelles comme nous le verrons au cours des prochains chapitres. Dans le Chapitre 3, nous présentons une méthode pour analyser et synthétiser deux grandes classes de textures structurelles : le carrelage rectangulaire et le bois. Nous verrons que, bien que notre méthode soit limitée à ces deux grandes classes de textures, elle fournit des résultats grandement supérieurs aux méthodes précédemment développées. Ensuite, dans le Chapitre 4, nous montrerons une méthode d'analyse et de synthèse de cartes de relief à partir de photographies. Le Chapitre 5 montre le résultat de nos textures appliquées sur différents modèles géométriques extraits à partir de photographies.



## Chapitre 3

# Analyse et synthèse de textures structurelles

*Le vieux charpentier trouve dans le poli de sa planche  
la récompense de son travail.*

*Citadelle, Antoine de Saint-Exupéry*

Avec l'apparition des techniques de modélisation à partir de photographies, il devient facile d'utiliser des textures extraites de la réalité pour ensuite les appliquer sur des mondes virtuels. Traditionnellement, les textures employées afin de donner une impression de complexité sont de simples images en couleur que l'on applique sur un modèle [Cat74, BN76]. Bien que cette méthode soit générale et simple à utiliser, elle présente de nombreux défauts : la résolution de la texture est fixe, l'illumination est souvent déjà incluse dans la texture (surtout si l'on utilise directement une texture provenant d'une ou de plusieurs photographies [DTM96, POF98]), les joints entre les faces du modèle géométrique sont souvent visibles, etc. Afin de répondre à certains de ces problèmes, les textures procédurales ont été introduites par Peachey [Pea85], Perlin [Per85], et Perlin et Hoffert [PH89]. Elles permettent de générer une texture à partir d'un algorithme qu'il faut bien souvent écrire à la main afin d'avoir l'effet voulu.

Les textures extraites à partir de photographies [DTM96, POF98] présentent souvent des "trous" qu'il faut remplir à l'aide de l'information présente autour de ce "trou" dans la texture. Pour ce faire de nombreux algorithmes d'analyse et de synthèse de textures ont été conçus [HB95, De 97, EL99, WL00] mais bien que ces algorithmes soient

généraux ils n'arrivent pas à bien synthétiser des textures structurelles (textures qui présentent une structure interne qui se répète régulièrement à travers la texture).

De nombreuses textures structurelles présentes dans les constructions humaines peuvent être paramétrées avec des textures procédurales. Ces textures procédurales offrent une puissante alternative à la technique d'application de textures traditionnelles. Par exemple, en ayant une procédure qui définit comment des briques sont agencées sur un mur il devient souvent possible d'assurer un joint parfait entre les différentes faces d'un modèle géométrique. Nous présentons dans ce chapitre, notre solution au problème de la génération des textures structurelles adaptées à une scène reconstruite à partir de photographies.

Le processus d'analyse et de synthèse peut être résumé par les étapes suivantes. Nous commençons par reconstruire la scène tridimensionnelle à l'aide de photographies et de notre logiciel de reconstruction à partir d'images [POF98]. Ensuite, les textures sont extraites des photographies et assignées à chacun des polygones formant notre scène. Pour chaque polygone et chaque texture qu'il veut générer, l'utilisateur sélectionne le type de texture structurelle (carrelage rectangulaire ou bois) et construit un masque binaire représentant la texture à l'aide de différents outils de traitement d'images [Umb98]. Étant donné la texture couleur, le masque binaire et la classe de la texture structurelle, le système analyse les caractéristiques de la texture et ajuste en conséquence les paramètres associés à cette classe de texture. Finalement l'algorithme procédural ainsi créé peut être utilisé pour générer des textures de haute résolution bien adaptées à la scène tridimensionnelle considérée. Les algorithmes d'extraction des paramètres sont complètement automatiques mais des interventions de l'utilisateur sont permises à toutes les étapes d'analyse afin de corriger ou de mieux orienter la recherche de certains paramètres.

Notre système peut créer des textures procédurales de qualité, libres des défauts généralement présents dans les textures extraites à partir de photographies. Étant donné que nous connaissons la structure interne de la texture générée, la texture procédurale résultant de nos analyses peut facilement être augmentée de différentes façons, incluant l'ajout d'un modèle de réflexion complet, d'une carte de micro-déplacement, d'une carte de relief, etc.

Dans les deux prochaines sections, nous détaillerons nos modèles procéduraux pour le carrelage rectangulaire et le bois. Nous avons choisi d'étudier ces deux classes de textures en particulier en raison de leur généralité et leur présence marquée dans presque toutes les constructions humaines. Dans le Chapitre 5 nous montrerons comment ces modèles peuvent être augmentés pour créer des textures plus sophistiquées.

Il est important ici de noter que la contribution du travail que nous détaillons dans ce chapitre est l'extraction automatique des paramètres pour les textures procédurales que nous présentons. Bien qu'on ne puisse citer aucun travail adressant spécifiquement la synthèse de carrelages ou de bois, il s'agit d'algorithmes simples et bien connus en général.

### 3.1 Le carrelage rectangulaire

Le premier modèle de texture que nous présentons est le carrelage rectangulaire. À cause de son omniprésence dans les constructions humaines, de nombreuses textures font partie de cette catégorie : les murs de briques, les revêtements de toiture et de murs extérieurs, les clôtures, les tuiles de céramique, les fenêtres de gratte-ciel, les plafonds suspendus, etc.

#### 3.1.1 Présentation du modèle

Notre modèle de carrelage est assez simple pour que l'on puisse extraire automatiquement ses paramètres à partir d'un masque binaire (l'extraction automatique des paramètres à partir de l'image originale non segmentée était impossible car les différentes composantes de la texture ne ressortaient pas assez) créé par l'utilisateur, qui représente la texture originale. Notre modèle, présenté en terminologie d'un mur de briques, est constitué de 6 paramètres scalaires (dépeints à la Figure 3.1) : longueur et hauteur des tuiles, orientation du motif, épaisseur du ciment, nombre de briques le long d'un mur, décalage relatif entre deux rangées de tuiles successives.

Afin de produire notre texture procédurale, nous commençons par segmenter la texture couleur (qui est une image de la texture en couleur et non segmentée produite par notre logiciel de reconstruction de scènes à partir de photographies) en un masque binaire qui sert à segmenter les différentes parties de la texture étudiée (par exemple

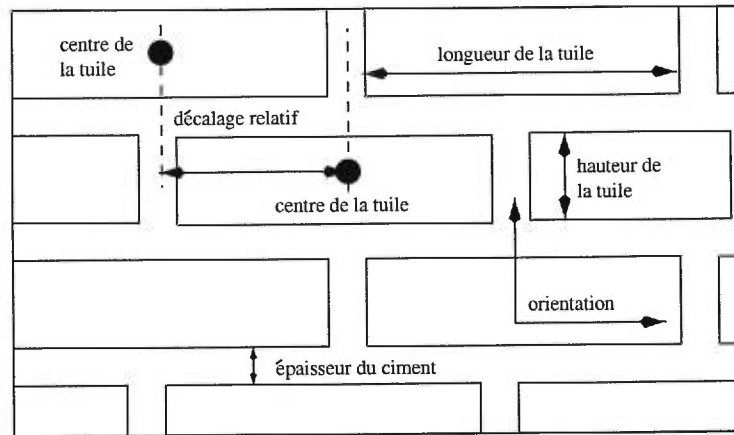


FIG. 3.1: Notre modèle de carrelage rectangulaire

dans le cas du carrelage rectangulaire le masque binaire distingue les briques du ciment) afin de faciliter l'extraction des paramètres dont nous avons besoin. Lors de cette étape, de nombreux filtres de segmentation dont un filtre de détection de contours de Sobel, un filtre pyramidal de détection de contours, un segmenteur d'histogramme et quelques opérateurs de morphologie mathématique (ouverture, fermeture, érosion), sont fournis à l'utilisateur afin de correctement segmenter et nettoyer la texture couleur qui sert d'entrée à notre algorithme. La plupart de ces outils sont fournis par la librairie de traitement d'images CVIP [Umb98]. Notons que cette étape de segmentation de la texture d'entrée est primordiale car certains de nos algorithmes d'extraction de paramètres peuvent être fortement influencés par le bruit présent dans le masque. Heureusement, l'utilisateur peut observer la qualité de son masque et le raffiner si nécessaire. Un exemple de masque extrait est présenté à la Figure 3.2. Normalement, un utilisateur habitué au système ne prendra pas plus de 1 ou 2 minutes afin de créer un masque binaire adéquat.

### 3.1.2 Extraction des paramètres du modèle

Dans cette section, nous discuterons de la façon d'extraire les paramètres de notre modèle de carrelage rectangulaire à partir du masque binaire fourni par l'utilisateur et de la texture d'entrée.

### Hauteur des tuiles et orientation du motif

Afin d'extraire ces deux paramètres, nous utilisons la transformée de Fourier. Nous avons choisi cette méthode en raison de sa robustesse par rapport au bruit présent dans le masque binaire qui segmente la texture à l'étude. Bien sûr, nous aurions aussi pu utiliser d'autres méthodes de transformation globales telles la transformée en cosinus et obtenir des résultats similaires. La transformée de Fourier a cependant produit des résultats satisfaisant pour notre application.

Pour bien situer le lecteur, nous allons maintenant décrire cette transformée et donner ses principaux attraits. Soit  $f(x)$  une fonction continue d'une variable réelle  $x$ , la transformée de Fourier notée  $F\{f(x)\}$  est définie par l'équation

$$F\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) \exp[-j2\pi ux] dx. \quad (3.1)$$

La variable  $u$  qui apparaît dans la transformée de Fourier est souvent appelée la *variable fréquentielle*. Ce nom vient de l'expression du terme exponentiel qui peut se décomposer en utilisant l'expression d'Euler en sinus et cosinus. En interprétant l'Équation 3.1 comme une limite de somme de termes discrets, il devient évident que  $F(u)$  est composée d'une somme infinie de sinus et cosinus et que chaque valeur de  $u$  détermine la fréquence de sa paire sinus/cosinus associée. En deux dimensions, la transformée de Fourier devient

$$F\{f(x, y)\} = F(u, v) = \iint_{-\infty}^{\infty} f(x, y) \exp[-j2\pi(ux + vy)] dx dy. \quad (3.2)$$

Dans notre cas, nous utilisons la version discrète de la transformée de Fourier en deux dimensions qui s'exprime comme

$$F\{f(x, y)\} = F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp\left[-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right] \quad (3.3)$$

pour  $x = 0, \dots, M - 1$   
 $y = 0, \dots, N - 1$ .

Notons ici que la transformée de Fourier est invariante sous la rotation. En effet, une rotation d'un angle  $\theta$  de l'image se traduit directement en la même rotation par  $\theta$  dans le domaine spectral (domaine de la transformée de Fourier). Pour s'en convaincre, le lecteur intéressé n'a qu'à substituer les variables  $x$  et  $y$  de l'Équation 3.2 par leurs équivalents en coordonnées polaires). Cette propriété est très importante pour nous car elle nous permet de calculer l'angle de rotation du motif de carrelage rectangulaire.

La transformée de Fourier (en fait la FFT pour *Fast Fourier Transform*) du masque binaire (voir Figures 3.2 et 3.3) donne d'importants indices en relation avec la hauteur des tuiles et l'orientation du motif.

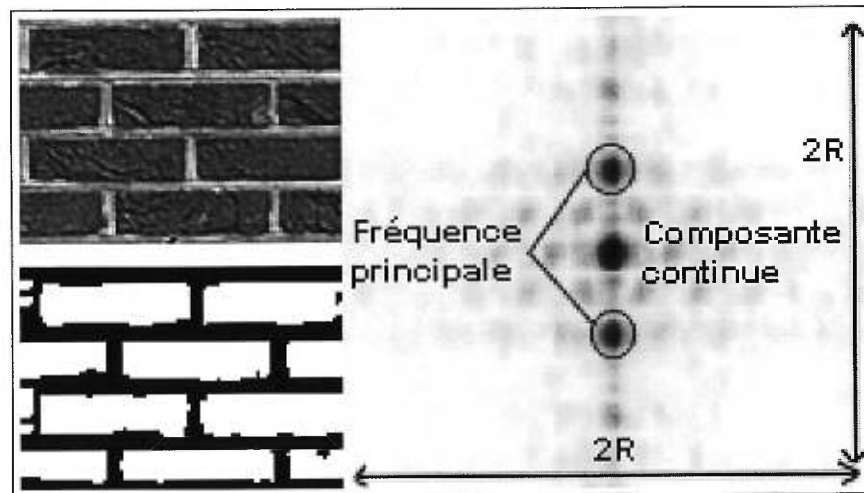


FIG. 3.2: Texture couleur, texture segmentée (masque binaire) et image négative et agrandie de la FFT du masque notée  $F(\hat{x}, \hat{y})$

Afin de bien illustrer la méthode, avant de calculer la transformée de Fourier de notre masque binaire, nous multiplions toute l'image  $f(x, y)$  par  $(-1)^{x+y}$  car ceci se traduit dans le domaine de Fourier en une translation qui amène la fréquence continue de l'image au centre de cette dernière (si cette opération n'est pas effectuée la composante continue se retrouve dans les quatre coins de l'image). Dans ce cas, pour une image de transformée de Fourier de résolution  $2R \times 2R$ , le centre du groupe de points le plus intense (ou le groupe de points le plus noir sur les images 3.2 et 3.3 dans notre cas car nous avons inversé les couleurs des FFTs afin de mieux voir les détails

suite à l'impression du mémoire)  $(\hat{x}, \hat{y})$  autre que le centre de l'image (la composante continue de l'image), correspond à la fréquence principale de la texture d'entrée. Dans un carrelage, cette fréquence est associée avec la répétition des lignes qui forment la base des rangées de tuiles. Si le carrelage est aligné, un autre groupe de points brillants représente les lignes verticales dans le motif et sera à  $90^\circ$  du premier groupe de points si l'on considère le centre de l'image comme pivot de rotation (voir Figure 3.3). En pratique, le point le plus brillant est trouvé dans la FFT en parcourant toute l'image et en ne retenant que les 10 points les plus brillants. Ensuite, l'entourage des points est étudié afin de savoir si les points font partie d'un agrégat (un groupe) ou non (seuls les points faisant partie d'un agrégat sont retenus car nous associons la présence d'un point seul à du bruit). Une fois les agrégats trouvés on prend le centre de l'agrégat le plus brillant en moyenne comme étant la fréquence principale de la texture.

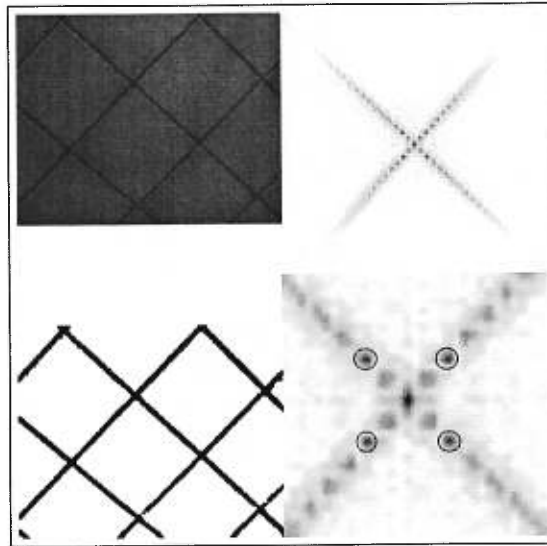


FIG. 3.3: Texture couleur, texture segmentée (masque binaire), négatif de la FFT du masque et agrandissement du négatif de la FFT

La relation entre le domaine spectral et le domaine spatial traduite par l'Équation 3.3 donne la hauteur des tuiles (mesurée comme une proportion de la hauteur totale de l'image) :

$$Hauteur = \frac{R}{2\sqrt{(\hat{x} - R)^2 + (\hat{y} - R)^2}}. \quad (3.4)$$

L'Équation 3.4 vient du fait que la transformée de Fourier est discrète. En effet, lorsque la transformée de Fourier est discrète, la distance entre deux échantillons dans le domaine spatial (donc la distance entre deux pixels de l'image originale) devient  $\frac{1}{2R}$  dans le domaine fréquentiel (domaine de la transformée de Fourier). Donc deux pixels adjacents dans la transformée de Fourier discrète ne sont pas séparés par un incrément de 1 mais bien par un incrément de  $\frac{1}{2R}$ , d'où l'Équation 3.4 qui traduit la distance entre la fréquence principale et la composante continue du domaine spectral vers le domaine spatial.

En vertu de la propriété de la transformée de Fourier mentionnée précédemment selon laquelle l'orientation des lignes qui forment la base des rangées de tuiles est invariante sous la transformée de Fourier, le point le plus brillant de la FFT  $(\hat{x}, \hat{y})$  donne aussi l'angle de rotation du motif de carrelage par

$$Angle = \arctan\left(\frac{\hat{x} - R}{\hat{y} - R}\right) \quad (\text{en radians}). \quad (3.5)$$

Le signe de l'angle de rotation est donné par le signe de  $(\hat{x} - R)$ . Ici, étant donné que la rotation est invariante sous la transformée de Fourier, on prend simplement l'angle entre la direction horizontale de l'image et la fréquence principale de l'image par rapport au centre de cette dernière.

La hauteur et l'orientation des tuiles sont extraites avec une très bonne précision car la FFT est une transformation globale qui résiste assez bien au bruit. Ces deux valeurs sont extraites en premier et servent de base pour l'extraction des autres paramètres.

Le motif formé par les lignes verticales de ciment est plus difficile à extraire à partir de la FFT lorsque les tuiles ne sont pas alignées (on peut facilement voir ce phénomène à la Figure 3.2 où la fréquence verticale majeure se perd à travers un nuage de points dans la FFT contrairement à la fréquence horizontale qui est clairement visible). Les prochaines sections décrivent l'approche spatiale que nous avons choisie



afin d'extraire les paramètres restants. Bien que du filtrage vertical de la FFT ou des techniques d'autocorrelation auraient pu être utiles, nos algorithmes se sont démontrés suffisamment stables et efficaces pour nos besoins.

### Longueur des tuiles et décalage relatif

Les paramètres suivants sont extraits en utilisant des algorithmes spatiaux plus précis que la transformée de Fourier. Il faut cependant noter que ces algorithmes sont plus sensibles au bruit. Pour simplifier les étapes suivantes, nous tournons le masque binaire de l'angle mesuré dans la section précédente dans le but d'aligner ce dernier avec les axes de la texture avant de faire l'extraction de nos paramètres.

Afin d'estimer la longueur des tuiles, nous parcourons un certain nombre de *lignes de balayage horizontales* en comptant le nombre de pixels blancs continus (tuiles) séparés par des pixels noirs (ciment). Les lignes de balayage horizontales se situant à l'intérieur du ciment sont ignorées ainsi que les tuiles non bornées par du ciment. Toutes les longueurs de tuiles sont ensuite triées et la médiane est prise. Le processus se répète quelques fois et la moyenne des médianes ainsi trouvée est prise comme étant la longueur de nos tuiles. Grâce à ce processus, les longueurs très différentes de la moyenne courante peuvent être facilement rejetées afin de réduire les effets du bruit dans le masque binaire.

En présence de motifs formés par des tuiles de longueurs différentes, cette approche peut être raffinée pour trouver la longueur de chaque type de tuile en séparant ces dernières en catégories. Nous n'avons pas approfondi cette direction d'étude mais les résultats en reconnaissance de formes [KMR72] pourraient être utilisés afin de déterminer des types de carrelages plus sophistiqués.

L'épaisseur du ciment est calculée de façon similaire, mais cette fois avec des lignes de balayage verticales. Cette orientation est moins affectée par le bruit dû à la segmentation. Si l'épaisseur horizontale est différente de l'épaisseur verticale, cette approche peut aisément être étendue pour satisfaire à un nouveau modèle.

Le décalage relatif entre deux rangées consécutives de tuiles est aussi extrait en utilisant des lignes de balayage horizontales. Une tuile est sélectionnée au hasard et sa longueur est estimée avec la méthode décrite au début de cette section. Une tuile adjacente sur une rangée au-dessus ou en-dessous de la rangée courante est trouvée en

parcourant une ligne de balayage verticale traversant une seule épaisseur de ciment. La longueur de cette tuile est aussi estimée comme pour la tuile précédente. Le décalage relatif est ensuite calculé à partir du centre des deux tuiles concernées. Le processus se répète pour plusieurs paires de tuiles adjacentes en mettant de côté les erreurs détectées et en prenant la moyenne des médianes des tuiles sélectionnées. Si le décalage relatif divisé par la longueur des tuiles est proche de zéro, nous présumons que nous sommes en présence d'un motif de tuiles alignées, ce qui est fréquent dans les carrelages de céramique.

### Les autres paramètres

Le nombre de tuiles le long d'un mur peut être extrait comme ci-haut. Cependant, il n'est pas toujours extrait automatiquement. Par exemple, nous utilisons souvent une vue rapprochée afin de mieux discerner les détails de la texture analysée (voir Figure 3.4).

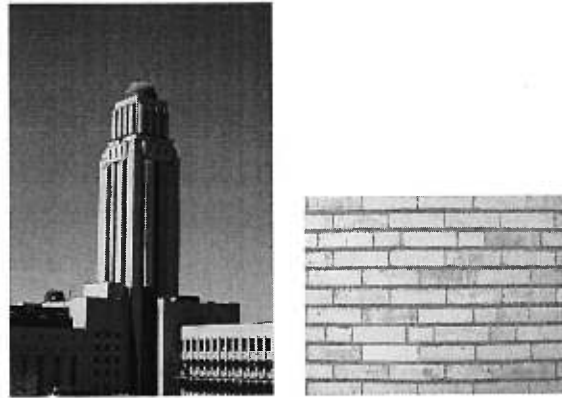


FIG. 3.4: Une photo de la tour de l'université de Montréal et une vue rapprochée qui montre les détails de la texture

Dans ce cas, le nombre de tuiles sur la vue rapprochée ne correspond pas du tout au nombre de tuiles le long du mur à reconstruire. De plus, de nombreuses photographies prises dans le but de reconstruire une scène tridimensionnelle, sont prises d'une certaine distance afin d'avoir une vue globale de l'objet à reconstruire. Souvent, à ces distances, le ciment devient invisible et l'on ne peut plus compter le nombre de tuiles (ou briques) le long d'un mur (voir Figure 3.4). Dans ces cas, l'utilisateur doit déterminer le nombre de tuiles présentes le long du mur qu'il considère et entrer manuellement cette

information dans le système. Une fois cette valeur détectée automatiquement ou entrée par l'utilisateur, le système peut disposer des tuiles de la bonne dimension sur le polygone tridimensionnel et conserver le rapport d'aspect de ces tuiles (c'est à dire avoir des tuiles de la même forme que sur la photographie).

La rugosité, la spécularité et la réflectance des tuiles ne sont pas extraites automatiquement. Pour le moment, l'utilisateur entre manuellement des valeurs dans le système afin d'obtenir les effets voulus. Dans le Chapitre 4 nous proposons une méthode d'extraction et de synthèse de cartes de relief (pour simuler la rugosité dans nos modèles) à partir d'une série de photographies.

### 3.1.3 Remplissage de l'intérieur des tuiles

L'intérieur des tuiles et du ciment est souvent de nature plus stochastique, et par le fait même plus approprié à une analyse et synthèse automatique par les algorithmes plus généraux [HB95, De 97, EL99]. Nous utilisons l'implantation en C de l'algorithme de Heeger et Bergen [HB95] faite par El-Maraghi [EM97] afin de générer l'intérieur de nos briques.

L'utilisateur sélectionne une ou plusieurs régions afin de représenter les tuiles, et une ou plusieurs régions pour représenter le ciment. Il devrait éviter les zones de texture qui présentent des défauts comme par exemple : des bloqueurs, des reflets, de l'ombre, une mauvaise résolution, etc. L'algorithme de Heeger et Bergen [HB95] produit ensuite automatiquement des textures similaires aux textures présentes dans les régions sélectionnées par l'utilisateur et ce, à la résolution voulue.

Parfois, les couleurs des tuiles d'un mur forment un motif ou les tuiles dans une certaine région peuvent être plus sales que les autres. Une fois tous les paramètres du carrelage extraits, il devient possible de décider à partir de quelle région sélectionnée on doit générer telle ou telle tuile en fonction de sa position et de la position des régions sélectionnées par l'utilisateur dans l'image originale. Notre système utilise une méthode semi-aléatoire afin de choisir l'échantillon de tuile réelle utilisé pour synthétiser une tuile donnée. Étant données la position du centre de chacun des  $N$  échantillons de tuiles réelles à l'intérieur de la texture originale et la position du centre de la tuile à générer en coordonnées de texture dans la texture à synthétiser, l'échantillon  $i$  sera

sélectionné selon l'algorithme 1.

---

**Algorithme 1:** Sélection d'un échantillon de tuile
 

---

```

Données :  $N$  : Le nombre d'échantillons de tuiles
              $Seuil$  : Seuil de distance fixé par l'utilisateur
              $S$  : Probabilité fixée par l'utilisateur
              $x$  : Distance centre à centre entre l'échantillon et la tuile à synthétiser
              $v$  : Échantillon le plus près de la tuile à synthétiser
              $s$  : variable probabiliste uniformément distribuée entre 0 et 1

Résultat :  $i$  : l'échantillon de tuile sélectionné

initialisation;
// Si un échantillon de tuile est assez près;
si  $x < Seuil$  alors
  // Calculer  $s$  la variable aléatoire;
   $s = DistributionUniforme()$ ;
  si  $s < S$  alors
     $i = v$ ;
  sinon
     $s = DistributionUniforme()$ ;
     $i = s \times (N - 1)$ ;
  fin
sinon
  // Sinon sélection aléatoire uniforme entre tous les échantillons;
  // Calculer  $s$  la variable aléatoire;
   $s = DistributionUniforme()$ ;
   $i = s \times (N - 1)$ ;
fin
retourner  $i$ ;

```

---

En d'autres termes, si la distance centre à centre entre la tuile à générer et l'échantillon de la tuile réelle la plus proche  $v$  est plus petite qu'un certain seuil, nous choisissons cet échantillon avec probabilité  $P(i = v) = S$  (plus la probabilité  $S$  est élevée, plus le motif original représenté par les tuiles réelles sera conservé). Sinon un échantillon est pris au hasard entre tous les échantillons disponibles. Cette façon de faire conserve la répartition originale des tuiles réelles tout en introduisant un certain facteur aléatoire afin de permettre à l'utilisateur de contrôler la ressemblance entre le motif formé par les tuiles synthétiques et celui formé par les tuiles réelles.

### 3.1.4 Quelques résultats du carrelage rectangulaire

Bien que les résultats seront présentés et discutés en détail dans le Chapitre 5, il convient de montrer quelques résultats préliminaires de cette méthode afin de situer le lecteur.

Les résultats présentés dans cette section utilisent notre méthode pour analyser et

synthétiser des carrelages rectangulaires. Tous les paramètres sont extraits automatiquement et les images présentées ne sont pas ré-illuminées, il s'agit de la texture à l'état brut.

Le premier exemple représente un mur de briques. La partie de gauche de la Figure 3.5 représente l'image originale. La portion centrale montre les résultats respectivement obtenus avec les algorithmes de DeBonet [De 97] et Efros et Leung [EL99] (meilleurs en général que Heeger et Bergen [HB95]). Enfin, la partie de droite donne le résultat obtenu avec notre algorithme. Bien que le résultat soit statistiquement semblable, il est facile de voir que la structure interne de la texture est perdue avec les algorithmes de DeBonet [De 97] et Efros et Leung [EL99] : les briques sont de tailles différentes, l'épaisseur du ciment varie beaucoup, il n'y a aucune séparation entre certaines briques, etc. Dans notre cas, à cause de la nature structurelle de notre modèle et des connaissances à priori que nous avons sur ce dernier, les briques sont de la même taille et le motif est conservé.

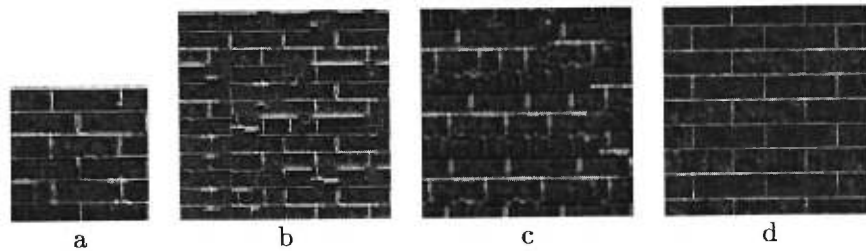


FIG. 3.5: Synthèse d'un mur de briques : (a) image originale; (b) synthèse de DeBonet [De 97]; (c) synthèse de Efros et Leung [EL99], implantation de Wei [Wei00]; (d) synthèse de notre algorithme

Le deuxième exemple montre qu'en sélectionnant les échantillons de briques et de ciment correctement, on peut facilement enlever certains défauts de l'image originale (dans ce cas le reflet au milieu de la Figure 3.6 à gauche).

Dans cet exemple, les échantillons de tuiles ont été choisis en périphérie de l'image originale afin d'éviter le reflet. De plus les échantillons de tuiles sélectionnés par l'utilisateur ont été répartis aléatoirement sur l'image synthétisée afin de simuler le carrelage de céramique réel (voir Figure 3.6). Notons aussi que malgré la courbure des lignes horizontales de ciment due à la déformation de lentille de l'appareil photographique, les paramètres du modèle sont bien approximés par nos algorithmes.

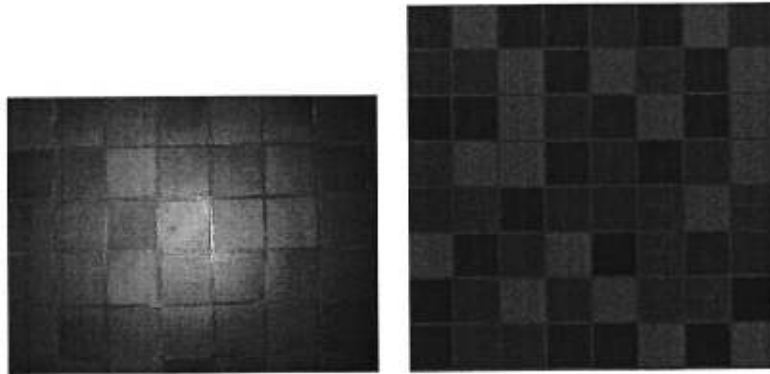


FIG. 3.6: Synthèse d'un plancher de céramique

## 3.2 Le bois

Les textures de bois, même si elles ne sont pas aussi fréquentes que le carrelage rectangulaire, représentent néanmoins une grande partie des textures dans notre environnement. Le bois est utilisé pour faire des planchers, portes, meubles, cadres de fenêtres, etc. Il peut être modélisé procéduralement à l'aide d'une fonction de croissance sinusoidale qui croît à partir du centre du tronc (voir Figure 3.7). Les paramètres du modèle peuvent être extraits à partir de photographies pour ensuite générer du bois sur des modèles tridimensionnels.

Une autre approche pour simuler le bois consiste à créer une texture anisotrope (texture qui présente une orientation dominante donnée comme des cheveux peignés par exemple) tridimensionnelle en utilisant une paire de photographies orthogonales [DGF98] et à utiliser cette texture volumique sur des modèles tridimensionnels. En faisant quelques hypothèses sur la structure interne d'une texture tridimensionnelle (la texture analysée doit correspondre plus ou moins à un champ aléatoire stationnaire sans macro-structures particulières visibles), Dischler *et al.* [DGF98] proposent une méthode efficace d'analyse et de synthèse basée sur une analyse hybride (spectrale et d'histogramme) d'une image 2D pour faire la synthèse de textures anisotropes tridimensionnelles.

Buchanan [Buc98] propose quant à lui une méthode de synthèse de bois basée sur le modèle de croissance biologique du bois. Plutôt que de simuler du bois en utilisant une texture procédurale, Buchanan [Buc98] propose de faire grandir un arbre dans une

matrice tridimensionnelle en suivant un modèle de croissance générique mais basé sur la réalité afin de pouvoir simuler les phénomènes de croissance locaux comme les noeuds.

Le modèle proposé par Buchanan [Buc98], bien que réaliste et complet, est assez complexe, requiert de nombreux paramètres et demande de grands temps de calcul car il faut simuler la croissance du bois. Dischler *et al.* [DGF98] n'ont pas vraiment de modèle quant à eux mais plutôt un champ aléatoire contraint qui est moins précis qu'un modèle procédural complet et ils ont besoin de deux vues orthogonales afin de créer le volume de texture.

Un aspect intéressant du modèle de bois que nous utilisons est que, à partir de valeurs extraites d'une seule image en deux dimensions, nous produisons une texture procédurale en trois dimensions. Elle a donc une résolution infinie et peut être appliquée sur n'importe quelle surface, incluant les surfaces courbées et les surfaces sans paramétrisation satisfaisante.

### 3.2.1 Présentation du modèle

Le tronc d'un arbre réel est constitué de cernes de croissance. Chaque année, un nouveau cerne de croissance s'ajoute sous l'écorce d'un arbre, dans la partie qu'on appelle l'aubier. C'est la partie active du tronc de l'arbre, celle qui achemine l'eau et les minéraux des racines aux feuilles. Au printemps, lorsque les conditions de croissance sont optimales, l'arbre produit des cellules plus grosses formant ainsi une couche plus poreuse appelée *bois de printemps* ou bois initial. À mesure que la saison avance, la croissance ralentit et les cellules produites deviennent plus denses pour former le *bois d'été* ou bois final. Ce sont les différentes largeurs et les nuances de couleurs des cernes de croissance qui donnent au bois son apparence unique et qui fait la beauté de nos tables, de nos bibliothèques et de nos parquets de bois.

Le tronc d'un arbre orienté selon l'axe  $Z$  sera donc modélisé par une série d'anneaux concentriques représentant les cernes de croissance de l'arbre avec alternance de couleur entre le bois de printemps et le bois d'été. Les couleurs de bois d'été et de printemps alternent de façon continue selon une distribution sinusoïdale de fréquence  $\omega$ , commençant à partir de l'axe central  $(x_c, y_c)$  du tronc. Il en résulte que la couleur de n'importe quel point tridimensionnel  $(x, y, z)$  peut être calculée grâce à l'expression

suivante (en pondérant les couleurs du bois d'été et de printemps par la valeur de l'expression) :

$$\sin \left( \omega \sqrt{(x - x_c)^2 + (y - y_c)^2} \right) + turbulence.$$

Une planche de bois peut être modélisée comme une coupe rectangulaire dans le tronc. Un angle d'inclinaison ainsi qu'un angle de rotation sont associés à la planche comme l'illustre la Figure 3.7. Nous avons choisi de garder notre modèle simple afin d'extraire tous ses paramètres à partir de photographies.

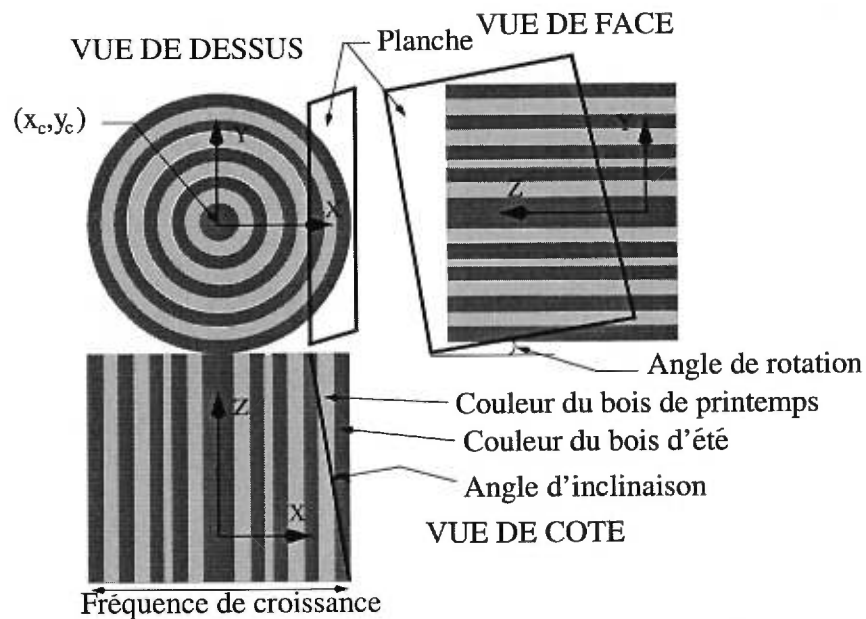


FIG. 3.7: Notre modèle de bois

Les 8 paramètres scalaires de notre modèle de bois sont suffisants pour reproduire une grande gamme de types de bois sans noeud ni écorce. Ces paramètres sont : la couleur du bois de printemps et la couleur du bois d'été, le centre du tronc  $(x_c, y_c)$ , l'angle d'inclinaison, l'angle de rotation, la fréquence de croissance, le ratio bois de printemps/d'été et l'intensité de la turbulence (forme de bruit continu introduite par Perlin [Per85]) présente dans le bois ainsi que sa fréquence.

Une fois de plus, l'utilisateur commence par convertir l'image d'entrée en un masque



binaire à l'aide d'opérateurs de segmentation et de détection de contours (voir Figure 3.8).

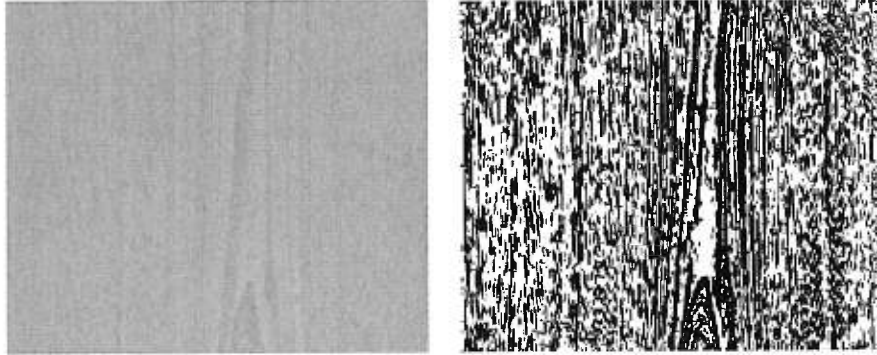


FIG. 3.8: Texture de bois et segmentation de cette texture

### 3.2.2 Extraction des paramètres du modèle

Dans cette section, nous discuterons de la façon d'extraire les paramètres de notre modèle de bois à partir du masque binaire fourni par l'utilisateur et de la texture d'entrée.

#### Couleurs du bois de printemps et du bois d'été, et ratio

Le masque binaire est traversé afin de déterminer le ratio entre le bois de printemps et le bois d'été. Dans cette section, le bois d'été est représenté par les pixels noirs et le bois de printemps par les pixels blancs.

Pendant que nous traversons les pixels correspondants dans la texture originale (en couleur), nous calculons la distribution des couleurs de chacun des deux types de bois en nous servant du masque binaire afin de déterminer le type de bois du pixel étudié (printemps ou été). Nous nous retrouvons donc avec deux distributions de couleurs, une pour le bois d'été et une pour le bois de printemps.

Nous avons observé que le 25<sup>ème</sup> percentile des couleurs du bois de printemps et le 75<sup>ème</sup> percentile des couleurs du bois d'été donnent des couleurs satisfaisantes. Cette solution empirique nous évite de délayer les couleurs fournies par la segmentation et est modifiable par l'utilisateur si nécessaire.

### Fréquence de croissance et angle de rotation

Le centre du groupe de points le plus brillant dans la transformée de Fourier (autre que la composante continue de l'image située au centre de la FFT) du masque binaire, correspond à la fréquence dominante de l'image segmentée. Cela nous donne la fréquence de croissance. L'orientation de cette fréquence dans le masque binaire détermine aussi l'angle de rotation de la planche par rapport au tronc. Les calculs se font grâce aux Équations 3.4 (en fait son inverse) et 3.5 de la Section 3.1.2 afin de déterminer la fréquence et l'orientation du bois. Évidemment, si le tronc d'arbre est vu du dessus (si le plan de coupe est le plan  $XY$ ), l'orientation de la fréquence peut donner n'importe quelle valeur entre  $0^\circ$  et  $360^\circ$ . Cette valeur sera cependant négligée car étant donné que seuls des cercles concentriques apparaîtront dans l'image, l'angle d'orientation du tronc est inutile.

Une fois l'orientation déterminée, le masque binaire et l'image sont tournés afin d'être alignés avec les axes de l'image pour l'extraction de l'angle d'inclinaison et des autres paramètres.

### Angle d'inclinaison

L'apparence de la texture de bois est hautement influencée par l'angle d'inclinaison de la planche par rapport au tronc. L'angle d'inclinaison contrôle principalement la forme des ellipses présentes dans la texture de bois. Pour cette raison, les ellipses présentes dans l'image originale donnent d'importants indices sur l'angle d'inclinaison à introduire dans notre modèle pour générer des effets similaires.

Notre modèle de bois ne tient pas compte de la présence de noeuds dans la texture. La détection et la synthèse de ces noeuds pourraient faire partie de travaux ultérieurs. Malheureusement, la présence de gros noeuds peut biaiser presque tous les algorithmes automatiques de détection car ces noeuds se comportent comme le tronc lui-même, mais à une plus petite échelle.

Notre solution courante est approximative, mais simple et efficace. Elle est basée sur l'observation que plus les ellipses courbent, de moins en moins de bois d'été (ou de moins en moins de bois de printemps dépendant de la position verticale dans l'image) apparaît entre des lignes de balayage horizontales successives (voir Figure 3.9). Si l'angle

d'inclinaison est nul, la texture montre seulement des lignes verticales identiques de bois de printemps et d'été et la variation entre deux lignes de balayage horizontales est donc nulle. Nous mesurons donc l'angle d'inclinaison en comptant le nombre de pixels de bois d'été dans des lignes de balayage successives et nous multiplions cette mesure par un facteur empirique. Cette mesure donne normalement de bons résultats lorsque l'angle d'inclinaison est proche de  $0^\circ$ , nous discuterons de ce point plus en détail à la Section 3.2.3.

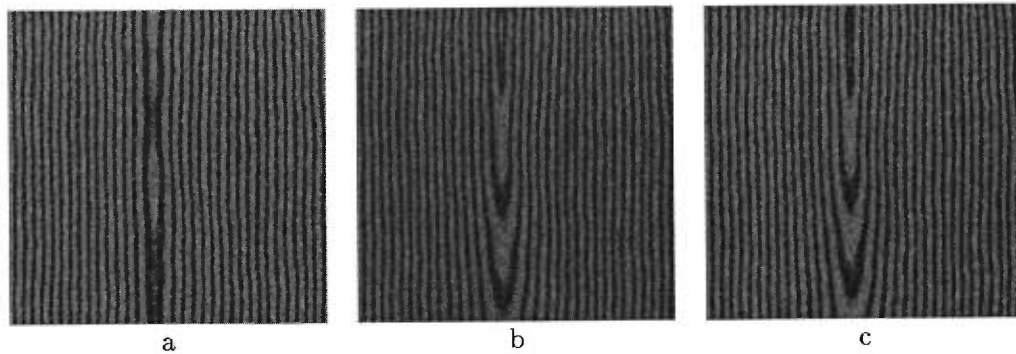


FIG. 3.9: Différence visuelle entre plusieurs valeurs d'angle d'inclinaison : (a)  $0^\circ$  ; (b)  $3^\circ$  ; (c)  $5^\circ$

Il est évident que cette méthode est sensible au bruit dans le masque et à la présence de noeuds dans l'image originale. Cependant elle a donné de très bons résultats visuels lors de nos tests.

Une autre façon d'extraire l'angle d'inclinaison de la planche pourrait être de suivre les ellipses dans l'image originale et d'extraire les paramètres de ces dernières (foyers, grand axe et petit axe) afin de déduire l'angle d'inclinaison à partir de la forme des ellipses. Nous n'avons pas étudié en profondeur cette méthode mais nous pensons qu'elle devrait donner de très bons résultats.

### Intensité de la turbulence et fréquence

Nous avons observé que la turbulence généralement présente dans le bois est habituellement de fréquence assez basse. Pour cette raison, nous utilisons une valeur typique et l'utilisateur peut la modifier si nécessaire.

L'intensité de la turbulence est estimée dans les régions de l'image où il y a peu

d'ellipses fortement courbées (loin du centre du tronc). Nous calculons la variation des types de bois (de printemps et d'été) entre les lignes verticales de pixels. S'il n'y a pas ou presque pas de variation, nous sommes en présence de lignes de bois parallèles et l'intensité de la turbulence est nulle. Sinon, cette intensité est fixée en rapport au degré de variation entre les lignes verticales. La Figure 3.10 illustre l'effet de la turbulence sur l'apparence du bois.

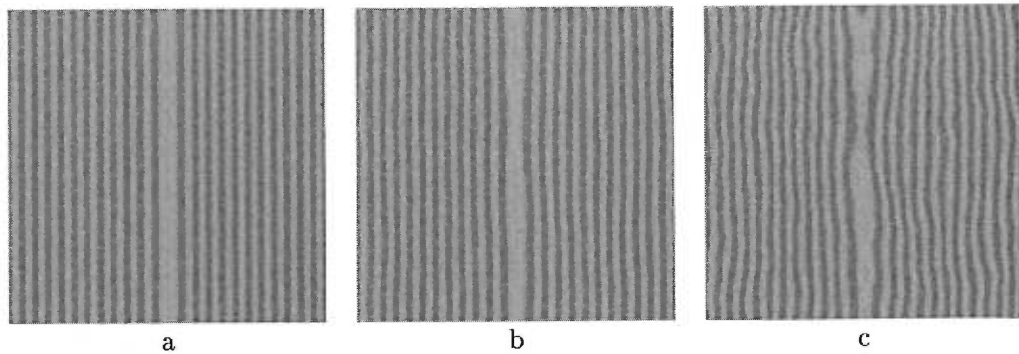


FIG. 3.10: Effet de la turbulence : (a) turbulence nulle; (b) turbulence faible; (c) turbulence moyenne

### Centre du tronc

Les coordonnées du centre du tronc  $(x_c, y_c)$  relativement à la planche sont aussi estimées statistiquement.

Supposons qu'une tangente à la planche est orientée vers  $Y$ . Si la planche est inclinée, des ellipses peuvent apparaître dans l'image. Le centre de ces ellipses correspond à la coordonnée  $Y$  ( $y_c$ ) du centre du tronc. Afin de trouver ce centre, nous calculons la variance entre les couleurs de chacune des colonnes de pixels de l'image. La colonne avec la plus grande variance (celle qui intersecte le plus d'ellipses) est associée avec le centre du tronc. Si la variance est plus petite qu'un certain seuil, une valeur de défaut en dehors de l'image est assignée à la coordonnée  $Y$ .

La coordonnée  $X$  est plus difficile à extraire. Pour un tronc positionné à  $x_c \neq 0$ , nous pouvons observer que la largeur de deux couches successives de types de bois varie comme l'illustre la Figure 3.11. Une fois de plus, des statistiques sur les rangées de pixels dans l'image sont utilisées pour indiquer la position de la coordonnée  $X$  ( $x_c$ )

du centre du tronc. Comme règle générale, si l'on ne considère pas la couche centrale qui peut être plus mince, plus on approche du centre du tronc, plus les couches augmentent en largeur. Encore une fois, un facteur empirique relie ces deux estimateurs statistiques aux valeurs réelles de la position du centre du tronc.

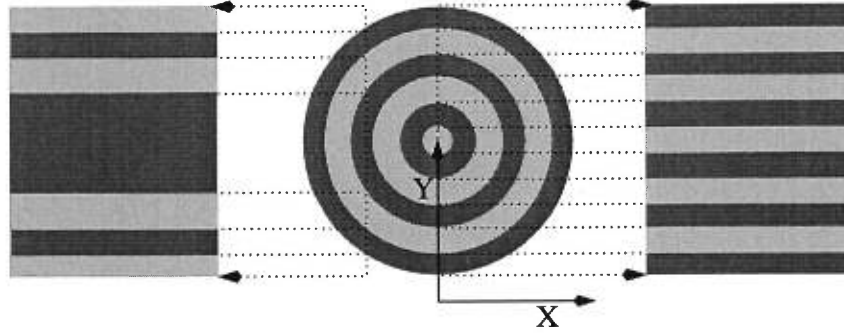


FIG. 3.11: Recherche de la coordonnée  $X$  du centre du tronc

### 3.2.3 Quelques résultats du bois

Même si nos algorithmes d'extraction de paramètres sont seulement approximatifs pour un bon nombre de paramètres, et que ces paramètres sont souvent inter-reliés, ils performant habituellement très bien quand on compare les résultats synthétisés avec des planches de bois réelles. La Figure 3.12 p. 40 montre un résultat typique de notre méthode.

Un aspect intéressant du modèle de bois que nous utilisons est que, à l'aide de paramètres extraits d'une seule image bidimensionnelle, nous générons une texture tridimensionnelle de résolution infinie qui peut être appliquée sur n'importe quelle surface, y compris les surfaces courbes.

Afin d'évaluer la précision de nos algorithmes d'extraction de paramètres, nous avons généré quelques textures de bois à l'aide de notre modèle procédural et nous avons comparé les valeurs des paramètres extraites par nos algorithmes avec les vraies valeurs. Nous avons généré dix sortes de bois pour lesquelles nous avons modifié manuellement tous les paramètres un à la fois. À la fin de chaque session nous avons calculé les différences entre les valeurs réelles et les valeurs estimées par nos algorithmes. Les résultats sont présentés au Tableau 3.1 comme des pourcentages de la différence

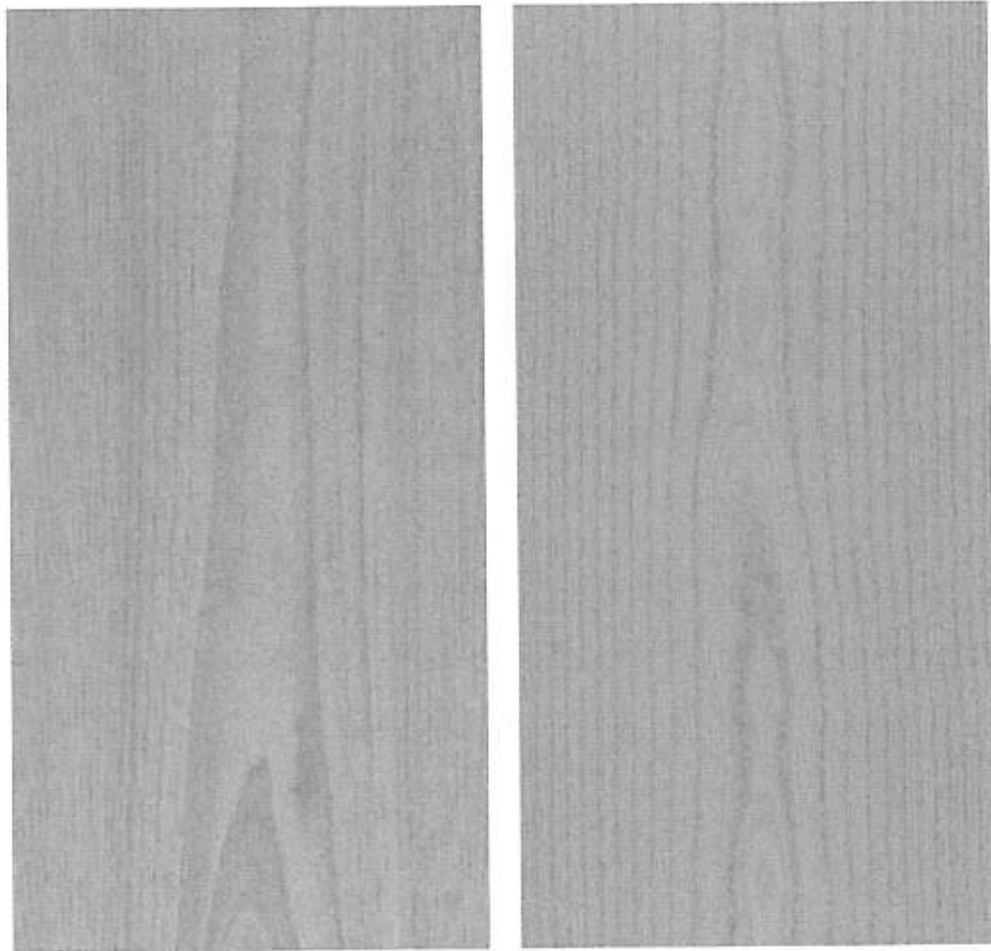


FIG. 3.12: Bois réel et bois synthétique

avec la valeur réelle. Notez que ces pourcentages sont calculés sur la valeur maximale de l'échelle pour la plupart des paramètres. Dans toutes les textures que nous avons générées, nous avons trouvé que l'erreur était proportionnelle à la valeur réelle pour l'angle d'inclinaison et la fréquence. Pour ces deux paramètres, nous avons donc calculé le pourcentage d'erreur en nous basant sur la valeur réelle plutôt que sur la valeur maximale de l'échelle (cette information est illustrée par le symbole \* dans le tableau).

Paramètres du modèle de bois	Valeur maximum de l'échelle	Statistiques sur les différences (%)		
		Moyenne	Écart-type	Maximum
Inclinaison	*°	8.09	4.56	14.29
Orientation	180°	0.35	0.24	0.6
Fréquence	*	2.47	4.81	12.34
Ratio bois de printemps/d'été	8	5.5	3.5	10.5
Intensité de la turbulence	30	10.3	7.16	20.63
$x_c$	512	2.71	2.46	6.56
$y_c$	512	2.27	2.21	6.50
Intensité du bois de printemps	256	3.18	0.76	4.94
Intensité du bois d'été	256	2.79	0.94	3.38

\* Proportionnel à la valeur réelle.

TAB. 3.1: Extraction des valeurs pour le bois simulé

Le paramètre qui a la moins bonne valeur estimée est l'intensité de la turbulence. Bien que le bois présente des altérations dues à la turbulence, la valeur exacte de cette dernière est peu importante. Nous avons observé que les valeurs que nous extrayons sont habituellement pleinement satisfaisantes. Étant donné que le ratio bois de printemps/d'été ainsi que la couleur du bois de printemps et du bois d'été sont des paramètres extraits à partir du masque binaire créé par l'utilisateur, les erreurs sont principalement causées dans ces cas par les légères différences entre les masques créés par l'utilisateur pour chaque texture.

Normalement, les planches de bois réelles sont coupées soit parallèlement ou perpendiculairement au grain du bois. En considération de ce fait, nous n'avons fait que de petites modifications à l'angle d'inclinaison autour de 0° et 90° afin d'évaluer l'estimation de ce paramètre par nos algorithmes. Notre algorithme d'extraction d'angle d'inclinaison est moins précis si la coupe est faite à un angle éloigné de 0° ou 90°. Par exemple, nous avons généré une texture de bois pour laquelle l'angle d'inclinaison était de 30° et la valeur extraite par notre algorithme était en erreur par 36% (ce résultat

ne figure pas au Tableau 3.1). Dans ces cas, étant donné que la forme des ellipses dans le bois est définie par l'angle d'inclinaison (voir Figure 3.9), l'utilisateur peut modifier la valeur de l'angle d'inclinaison afin d'obtenir l'effet voulu.

Pour le reste des paramètres, nos algorithmes sont normalement suffisamment précis avec des erreurs autour de 2 à 5% (voir Tableau 3.1). Dans le cas d'une erreur fortuite, l'utilisateur peut corriger le problème en quelques étapes. Par exemple, pour le paramètre  $y_c$ , l'utilisateur peut facilement spécifier le centre des ellipses dans l'image originale.

Nous avons présenté dans ce chapitre une méthode d'analyse et de synthèse de deux grandes classes de textures structurelles : le carrelage rectangulaire et le bois. Bien que nos méthodes d'analyse et de synthèse sont spécifiques à ces deux classes de textures, nous pensons qu'il serait relativement facile d'intégrer d'autres types de textures structurelles et que les outils que nous avons développés pour extraire les paramètres de nos textures pourraient servir à extraire les paramètres d'un nouveau type de texture structurelle.



## Chapitre 4

# Extraction et synthèse de cartes de relief

Dans le chapitre précédent, nous avons présenté une méthode d'analyse et de synthèse de deux grandes classes de textures structurelles : le carrelage rectangulaire et le bois. Grâce à cette méthode, nous avons une définition procédurale de la texture que nous voulons afficher. Il serait maintenant intéressant de pouvoir augmenter le réalisme de la texture synthétique en ajoutant un relief cohérent sur cette dernière. Par exemple on peut vouloir que les côtés de tuiles de céramique soient courbés vers le ciment et que la surface de chaque tuile soit légèrement ondulée, comme c'est le cas pour de vraies tuiles de céramique. Dans cette optique, nous explorons dans ce chapitre une méthode d'extraction et de synthèse de cartes de relief dans le but de simuler un relief semblable à celui présent sur les objets réels.

Comme nous le verrons à la Section 4.2, beaucoup de travail a été fait en traitement d'image et en vision afin d'extraire une carte de relief à partir de surfaces diffuses. Malheureusement, avec ce type de surfaces, les détails fins de la surface ne sont pas visibles et ne peuvent donc être extraits. Nous proposons donc dans ce chapitre d'utiliser le reflet spéculaire d'une surface afin d'extraire les fins détails de relief d'une surface et de stocker ces derniers dans une carte de relief.

Tout d'abord, afin de situer le lecteur, nous allons présenter quelques notions préalables nécessaires à la compréhension du chapitre, ensuite nous présenterons les travaux antérieurs dans le domaine et nous détaillerons nos travaux.

## 4.1 Notions préalables

Nous présentons dans cette section les notions préalables à la compréhension de ce chapitre. Nous commencerons cette section en expliquant ce qu'est une *BRDF*. Nous poursuivrons en définissant le modèle d'illumination simple (le modèle de Phong) utilisé dans ce chapitre. Ensuite, nous expliquerons comment simuler un relief sur une surface en utilisant une carte de relief.

### 4.1.1 La BRDF

La BRDF (*bidirectional reflectance distribution function*) d'une surface est une fonction qui décrit comment une surface interagit avec la lumière [War92]. Plus précisément, la BRDF est définie comme étant le ratio de la radiance réfléchie dans la direction  $\omega_r$  étant donnée l'irradiance différentielle provenant d'une direction incidente  $\omega_i$ . Une BRDF se définit mathématiquement comme suit

$$f_r(\omega_i, \omega_r) \equiv \frac{L_r(\omega_r)}{L_i(\omega_i) \cos \theta_i d\omega_i} \quad (4.1)$$

avec

- $L_i(\omega_i)$  = irradiance provenant de la direction  $\omega_i$
- $L_r(\omega_r)$  = radiance réfléchie dans la direction  $\omega_r$
- $d\omega_i$  = angle solide soutendu par  $\omega_i$
- $\theta_i$  = angle entre la normale de la surface et  $\omega_i$ .

La Figure 4.1 illustre la BRDF et les variables impliquées dans l'Équation 4.1 (afin de simplifier l'Équation 4.1 et la Figure 4.1, la longueur d'onde de la lumière est négligée dans ces dernières).

L'Équation 4.1 est l'expression analytique la plus complète d'une BRDF. Bien souvent, certaines hypothèses quant aux propriétés de la surface sont posées afin de simplifier le problème. La BRDF d'une surface peut être représentée sous la forme d'un tableau à 7 dimensions ( $\theta_i, \phi_i, \theta_r, \phi_r, L_i, L_r, \lambda$ ) qui échantillonne les valeurs de radiance et d'irradiance sous différents angles et longueurs d'onde de la lumière. On peut aussi approximer une BRDF en utilisant un modèle mathématique plus ou moins compliqué afin d'éviter d'avoir à garder en mémoire toutes les informations relatives à la BRDF d'une surface. Le modèle de Phong [Pho75] est un exemple d'approximation d'une

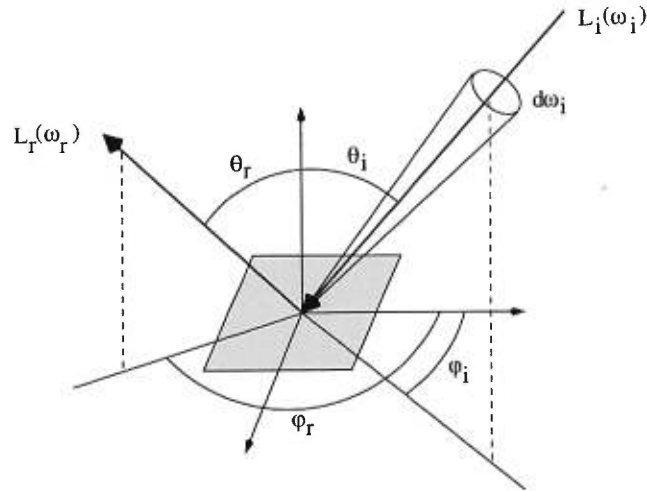


FIG. 4.1: BRDF

BRDF isotrope (qui ne dépend pas de  $\phi$ ) par un modèle mathématique simple mais non physiquement valable.

#### 4.1.2 Le modèle de Phong

Le modèle de réflexion de Phong [Pho75] est très souvent utilisé en infographie pour approximer les BRDFs isotropes car il est à la fois simple et rapide à évaluer. De plus, encore de nos jours, il reste un des seuls modèles d'illumination à être implanté sur du matériel graphique de façon répandue. Nous allons utiliser ce modèle d'illumination afin d'évaluer les BRDFs des surfaces que nous allons étudier en raison de sa simplicité. Le modèle de Phong est défini par l'Équation 4.2 :

$$L_r(\omega_r) = I_a k_a + I_d k_d (\mathbf{N} \cdot \mathbf{L}) + I_s k_s (\mathbf{R} \cdot \mathbf{V})^n \quad (4.2)$$

avec

- $L_r(\omega_r)$  = Radiance réfléchie dans la direction  $\omega_r$
- $I_a, I_d$  et  $I_s$  = Intensité lumineuse ambiante, diffuse et spéculaire
- $k_a, k_d$  et  $k_s$  = Coefficients ambiant, diffus et spéculaire de la surface
- $n$  = Coefficient de rugosité
- $\mathbf{N}$  = Normale à la surface
- $\mathbf{L}$  = Direction de la lumière incidente
- $\mathbf{R}$  = Direction réfléchie de la lumière
- $\mathbf{V}$  = Direction de l'observateur (point de vue).

La Figure 4.2 illustre comment le modèle de Phong [Pho75] approxime une BRDF isotrope.

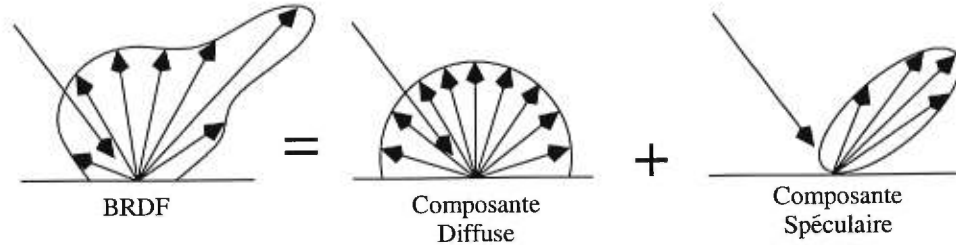


FIG. 4.2: Modèle d'illumination locale de Phong

### 4.1.3 La carte de relief

Comme nous avons pu le voir à l'Équation 4.2, la normale de la surface et l'indice de rugosité  $n$  influent grandement l'apparence d'une surface éclairée par une ou plusieurs lumières. Un autre facteur joue un rôle important dans l'apparence visuelle d'une surface : la micro-géométrie de cette dernière. Afin de simuler un relief sur une surface, une méthode consiste à modifier la normale de la surface de façon à perturber la façon dont la lumière interagira avec la surface et ainsi créer le relief voulu. Les informations concernant la perturbation des normales à la surface sont stockées dans une carte de relief. Nous allons utiliser cette façon de faire pour simuler la rugosité des surfaces que nous étudions car cette méthode est peu coûteuse en calcul et donne des résultats visuels très satisfaisants.

## 4.2 Travaux antérieurs en analyse et synthèse de cartes de relief et de BRDFs

Nous présentons sommairement dans cette section les travaux antérieurs touchant l'analyse et la synthèse de cartes de relief et des BRDFs. Nous allons maintenant présenter les travaux que nous jugeons les plus pertinents.

Les deux premières méthodes que nous présentons extraient les paramètres d'une BRDF sous une illumination locale et calibrée.

Ward [War92] extrait une BRDF à partir d'échantillons réels. À l'aide d'un gonioréfectomètre (appareil servant à mesurer la radiance d'une surface) fait maison et constitué d'une caméra et d'une sphère diffuse, Ward échantillonne la BRDF d'une surface et simule cette BRDF de façon synthétique en ajustant un modèle de réflectance aux échantillons prélevés par son gonioréfectomètre. Le modèle proposé par Ward est anisotrope et est une extension du modèle Gaussien de Torrance et Sparrow [TS67].

Sato *et al.* [SWI97] utilisent un robot afin d'échantillonner un objet sous différents angles afin d'extraire la BRDF de ce dernier. La reconstruction géométrique de l'objet étudié est faite en combinant plusieurs cartes de profondeur. La composante diffuse est séparée de la composante spéculaire en projetant les images couleur provenant du robot sur le modèle tridimensionnel. Ensuite, le modèle de réflexion de Torrance et Sparrow [TS67] est ajusté sur les échantillons prélevés par le robot et la BRDF est ainsi estimée. La méthode a l'avantage d'extraire tous les paramètres d'une surface (BRDF et relief) mais demande un environnement hautement calibré et des appareils sophistiqués.

Lorsque la surface à partir de laquelle nous voulons extraire une BRDF ou une carte de relief est placée dans un environnement réel, il peut être important de tenir compte des effets de l'illumination indirecte présente dans la scène. Les deux prochaines méthodes présentées extraient le relief de surfaces en tenant compte de cette illumination indirecte.

Loscos *et al.* [LFD<sup>+</sup>99] proposent une méthode d'extraction de BRDFs de surfaces en vue d'effectuer le ré-éclairage virtuel de scènes réelles. Pour ce faire, un modèle géométrique est créé à partir de photographies en utilisant un logiciel de photogrammétrie [POF98]. Ensuite, en utilisant les informations contenues dans les photographies et le modèle géométrique, un modèle diffus simple est ajusté sur chacune des surfaces pour trouver la réflectance de chaque surface.

Yu *et al.* [YDMH99] proposent une façon de récupérer les modèles de réflectance (BRDFs) de scènes réelles à partir de photographies. L'algorithme fonctionne à l'aide d'un modèle créé par modélisation à partir de photographies à l'aide du système *Facade* [DTM96] et de cartes de radiance de ces photographies créées selon l'algorithme de Debevec et Malik [DM97]. Étant données ces entrées, on partitionne la scène de façon hiérarchique en un maillage de polygones et on utilise le rendu basé sur des images pour

calculer à la fois la radiance et l'irradiance de chaque polygone à partir de l'information contenue dans les cartes de radiance. Par la suite, la position supposée des reflets spéculaires est calculée (étant donnée la position des lumières et des polygones) et la BRDF de la surface (coefficients spéculaire et diffus) est calculée en utilisant le modèle de Ward [War92] dans un processus d'optimisation itératif. Évidemment, si aucun reflet spéculaire n'est détecté, la surface est supposée parfaitement diffuse.

Enfin, nous présentons la méthode qui se rapproche (mais qui utilise l'illumination diffuse plutôt que spéculaire) le plus de la méthode que nous proposons et qui sera décrite à la prochaine section.

Rushmeier *et al.* [RTG97] proposent une méthode pour trouver une carte de relief d'une surface à partir d'un objet réel sous un nombre limité de sources de lumière calibrées. Le système est basé sur le fait que l'on peut déduire la forme d'un objet presque complètement diffus grâce à la différence de l'interpolation de l'illumination sur l'objet étudié [ZZ88, Pen88, Wei90]. La méthode fonctionne même si l'objet n'est pas de couleur uniforme mais ne marchera pas avec des objets ayant des crevasses profondes ou une carte de relief de très faible intensité. De plus, les réflexions spéculaires peuvent introduire des erreurs importantes lors de l'évaluation de la carte de relief.

### 4.3 Analyse et synthèse de cartes de relief

L'ajout de cartes de relief à une texture produit un effet beaucoup plus réaliste qu'une simple texture couleur. Comme nous l'avons mentionné dans le Chapitre 3, la carte de relief de nos textures structurelles sont entrées à la main par l'utilisateur.

Dans ce chapitre, nous proposons une méthode pour extraire une carte de relief à partir de photographies d'une surface planaire spéculaire. Nous avons choisi d'extraire le relief d'une surface à partir d'une réflexion spéculaire plutôt que d'utiliser l'illumination diffuse car les détails fins dans le relief sont mieux perçus dans les zones spéculaires. En effet, dans les zones spéculaires (là où le troisième terme de l'Équation 4.2 est grand), à cause de l'exposant  $n$  de l'équation de Phong (Équation 4.2), une fine variation de normale entraînera un grand changement dans l'illumination.

L'extraction de la carte de relief se fait de façon interactive à partir d'une carte de radiance créée grâce à la méthode proposée par Debevec et Malik [DM97] (qui sera

décrite à la Section 4.3.1). Ensuite, nous utilisons l'algorithme de Efros et Leung [EL99] afin de synthétiser une carte de relief de la grandeur voulue. Cette section se divise donc en trois grandes parties soient l'extraction de la carte de relief à partir d'une carte de radiance, la synthèse de la carte de relief et enfin, le rendu.

### 4.3.1 Extraction de la carte de relief

Étant donné que les réflexions spéculaires (terme spéculaire dans l'Équation 4.2) sont très sensibles aux perturbations des normales de surface, les variations fines d'intensité dues à des variations de hauteur de la surface peuvent être plus facilement perçues sur des surfaces hautement spéculaires que sur des surfaces diffuses. Ce constat étant posé, nous supposons aussi que :

1. La surface est hautement spéculaire (la surface va causer des reflets)
2. La direction de l'appareil photographique et de la lumière sont presque parallèles
3. La surface est presque perpendiculaire à la lumière (et à l'appareil photographique)
4. La surface est plane
5. La lumière est ponctuelle (ou la taille de la lumière est négligeable)
6. Une seule lumière éclaire la surface.

Sous de telles conditions, un pixel clair implique que la normale de la surface est perpendiculaire à la lumière (et à la caméra), et un pixel foncé implique que la normale de la surface n'est pas perpendiculaire ou est dans une région d'ombre.

### Extraction de cartes de radiance à partir de photographies

Les réflexions spéculaires dans les photographies digitales sursaturent souvent l'image causant ainsi une perte d'information. Afin d'utiliser correctement l'information contenue dans une réflexion spéculaire, nous avons besoin d'une source d'information plus précise qu'une simple photographie digitale. Debevec et Malik [DM97] proposent pour retrouver la fonction de transfert d'un film (fonction qui décrit comment le film réagit à la lumière) d'utiliser l'Équation (4.3) qui simule la courbe de réponse d'un film à la lumière.

$$\begin{aligned}
\mathcal{O} = & \sum_{i=0}^N \sum_{j=0}^P \{w(Z_{ij}) [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 \\
& + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2
\end{aligned} \tag{4.3}$$

où	$i$	est l'éventail des pixels d'une image
	$j$	est l'éventail des photographies
	$P$	est le nombre de photographies
	$N$	est le nombre de pixels d'une image
	$Z_{ij}$	est la couleur de l'image $j$ en position $i$
	$Z_{min}$ et $Z_{max}$	sont respectivement les couleurs minimales et maximales
	$w()$	est la fonction de poids pour assurer une courbe $g()$ plus lisse
	$g()$	est la fonction de réponse du film (à trouver)
	$E_i$	est la radiance de la scène au point $i$ (à trouver)
	$\lambda$	est une constante qui détermine le lissage de la courbe
	$\Delta t_j$	est le temps d'exposition de la photographie $j$ .

Dans l'Équation 4.3, on cherche à trouver une valeur à  $g()$  et à  $E_i$  qui minimisent l'équation au sens des moindres carrés.

Une fois cette équation résolue, il devient facile de composer une série de photographies prises à différentes expositions en une carte de radiance non-saturée. Nous utilisons cette méthode afin d'enlever la sursaturation dans les réflexions spéculaires.

### 4.3.2 Le montage

Afin de simuler les conditions requises par nos hypothèses de travail, nous avons effectué le montage présenté à la Figure 4.3.

Dans ce montage, on place une puissante lumière halogène (afin d'avoir une lumière blanche) derrière un carton opaque que l'on a perforé afin de simuler une lumière ponctuelle. L'appareil photographique est placé en-dessous du trou d'épingle afin d'être le plus parallèle possible à la lumière. La surface à étudier est maintenue verticalement afin d'être perpendiculaire à la lumière.

### 4.3.3 Approximation par le modèle de Phong

Étant donnée la carte de radiance, l'utilisateur indique une région comprenant le reflet



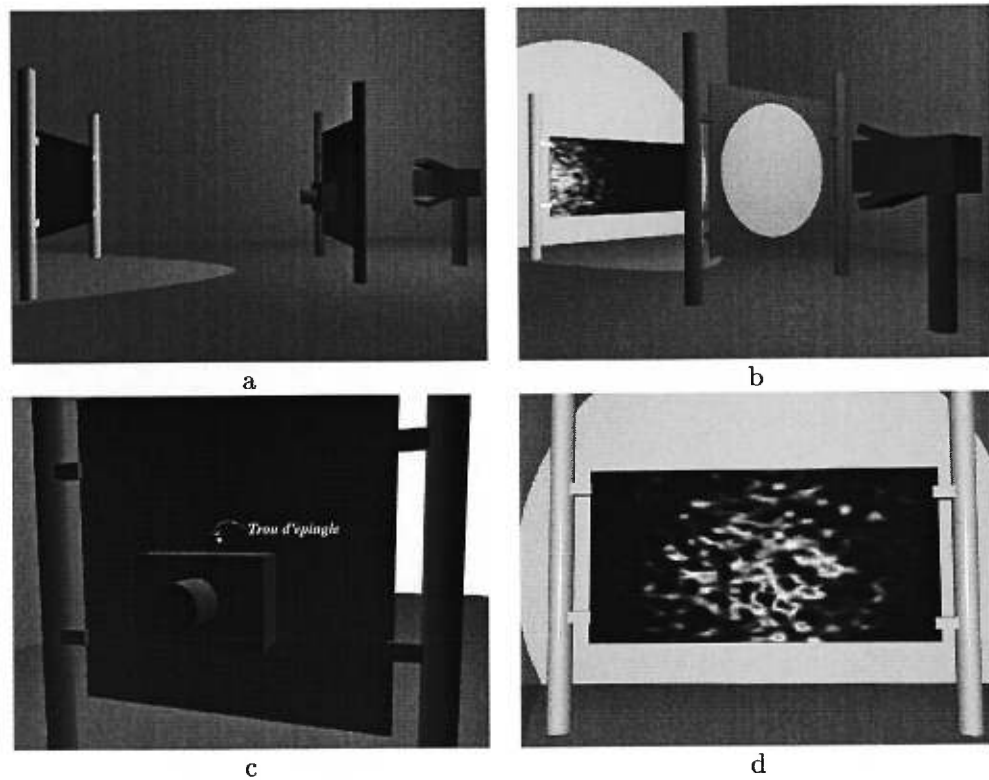


FIG. 4.3: Le montage : (a) vue de côté ; (b) vue arrière ; (c) vue présentant l'appareil photographique ainsi que le trou d'épingle ; (d) vue de l'appareil photographique

spéculaire qui nous intéresse. Ensuite, nous approximations le reflet spéculaire (troisième terme de l'Équation 4.2) à l'aide du modèle de réflexion spéculaire de Phong [Pho75] défini par l'équation 4.4.

$$I_{spec} = I k_s \cos^n \alpha \quad (4.4)$$

avec

- $I_{spec}$  = Intensité de la réflexion
- $k_s$  = Coefficient spéculaire
- $n$  = Exposant de rugosité
- $\alpha$  = Angle entre la direction de la lumière réfléchie et la direction de vue
- $I$  = Intensité de la lumière.

L'approximation se fait en utilisant une recherche dichotomique afin de trouver l'exposant de rugosité ( $n$ ) qui minimise la somme des différences entre le modèle de Phong et les valeurs présentes dans la carte de radiance. Les autres paramètres de l'Équation 4.2 sont supposés nuls étant donné qu'au point de la réflexion spéculaire le troisième terme de l'Équation 4.2 est nettement plus grand que les autres. Le processus d'approximation peut se séparer en deux grandes étapes : trouver le centre de la réflexion spéculaire dans la carte de radiance (Figure 4.4(b)) et approximer le coefficient de rugosité (Figure 4.4(c)). Afin de trouver le centre de la réflexion nous utilisons la valeur de l'intensité dans la carte de radiance pondérée par sa position  $(x, y)$ . L'algorithme 2 présente en détail ces deux étapes.

La dernière étape de ce processus consiste à soustraire les valeurs dictées par modèle de Phong [Pho75] aux valeurs contenues dans la carte de radiance afin d'enlever les effets de la réflexion spéculaire. Malheureusement, l'approximation par le modèle de Phong peut ne pas être parfaite (étant donné que le modèle est très simple) et quelques effets de la réflexion spéculaire peuvent être encore présents dans la carte de radiance (Figure 4.4(d)). Afin d'enlever ces effets indésirables, nous utilisons la transformée de Fourier (en fait la FFT) de l'image de radiance soustraite du modèle de Phong et nous la filtrons à l'aide d'un filtre passe-haut. Le fait de filtrer l'image n'enlève pas les détails importants pour nous car nous cherchons à extraire les reliefs fins de la carte de radiance. Ces détails fins sont généralement de haute fréquence et ne seront donc pas filtrés. Ce procédé nous donne une carte de radiance épurée des effets de la réflexion qui

**Algorithme 2:** Processus d'approximation

---

**Données :** *Radiance* : Tableau contenant la partie de la carte de radiance sélectionnée par l'utilisateur  
*Somme* : La somme des valeurs de la ligne en cours  
*SommeX* : La somme des valeurs de la ligne en cours pondérée par la coordonnée *X* de la valeur (en pixels)  
*SommeY* : La somme des valeurs de la ligne en cours pondérée par la coordonnée *Y* de la valeur (en pixels)  
*CentreX* : Coordonnée *X* du centre de la réflexion spéculaire  
*CentreY* : Coordonnée *Y* du centre de la réflexion spéculaire  
*Max* : Valeur maximale de la réflexion spéculaire  
*nAvant* : Borne inférieure de *n* courante (dans la recherche dichotomique)  
*nAprès* : Borne supérieure de *n* courante (dans la recherche dichotomique)  
*Différence* : Différence entre la réflexion spéculaire et le modèle de Phong  
*Phong* : Tableau contenant les valeurs du modèle de Phong

**Résultat :** *n* : la valeur de l'exposant de rugosité

```
// Trouver le centre de la réflexion;
Somme = SommeX = SommeY = Max = 0;
pour j < Radiance → Largeur faire
  pour i < Radiance → Hauteur faire
    Somme += Radiance[i][j];
    SommeX += Radiance[i][j] * i;
    SommeY += Radiance[i][j] * j;
    si Radiance[i][j] > Max alors
      Max = Radiance[i][j];
    fin
  fin
fin
CentreX = SommeX / Somme;
CentreY = SommeY / Somme;

// Trouver n;
n = 5000;
tant que Différence trop grande faire
  // Initialisation du tableau de Phong pour la rugosité n;
  CalculerPhong(Phong, n, Max);
  Différence = 0;
  nAvant = 0;
  nAprès = 25000;
  pour j < Radiance → Largeur faire
    pour i < Radiance → Hauteur faire
      Différence += Radiance[i][j] - Phong[i][j];
    fin
  fin
  si Différence > 0 alors
    nAprès = n;
    n = n - (n - nAvant) / 2;
  sinon
    nAvant = n;
    n = n + (nAprès - n) / 2;
  fin
fin
retourner n;
```

---

sert de base pour la synthèse de la carte de relief (Figure 4.4(e)). Cette carte de radiance épurée peut maintenant être considérée comme une carte de relief à un facteur d'échelle près car les différences d'intensités entre les pixels de la carte de radiance sont dus aux micro-détails de la surface. La Figure 4.4 illustre chacune des étapes du processus.

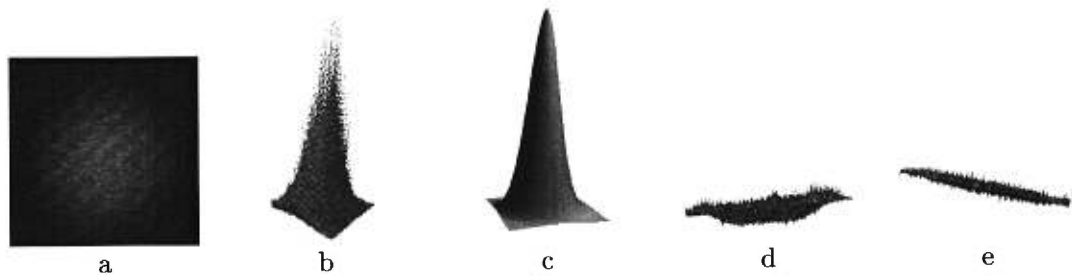


FIG. 4.4: (a) Image d'une réflexion spéculaire; (b) vue en perspective d'élévation de la carte de radiance reconstruite; (c) modèle de Phong [Pho75] approximant la carte de radiance; (d) carte de radiance après la soustraction des valeurs dictées par le modèle de Phong [Pho75]; (e) carte de radiance après la soustraction et filtrage passe-haut

#### 4.3.4 Synthèse de la carte de relief

Afin de synthétiser la carte de relief, nous utilisons l'information à l'intérieur de la réflexion spéculaire contenue dans la carte de radiance résultant des étapes précédentes. Nous utilisons l'implantation de Wei [Wei00] de l'algorithme de Efros et Leung [EL99]. Cet algorithme prend un temps considérable (environ 48 heures sur un Athlon 500 Mhz pour une image  $512 \times 512$ ) à synthétiser une texture mais donne les meilleurs résultats observés en présence de textures semi-structurées (Wei et Levoy [WL00] ont amélioré grandement la vitesse de cet algorithme). La Figure 4.5 donne un exemple de synthèse de carte de relief.

#### 4.3.5 Rendu et résultats

Nous avons produit nos exemples en utilisant un algorithme de lancer de rayons standard avec 10 rayons par pixel pour réduire l'aliasage important dû aux hautes fréquences présentes dans les images à cause des détails fins de la surface. L'exposant  $n$  de la réflexion spéculaire ainsi que la carte de relief ont été calculés avec la méthode que nous venons de présenter. Pour le moment, l'utilisateur doit donner le facteur d'échelle qui

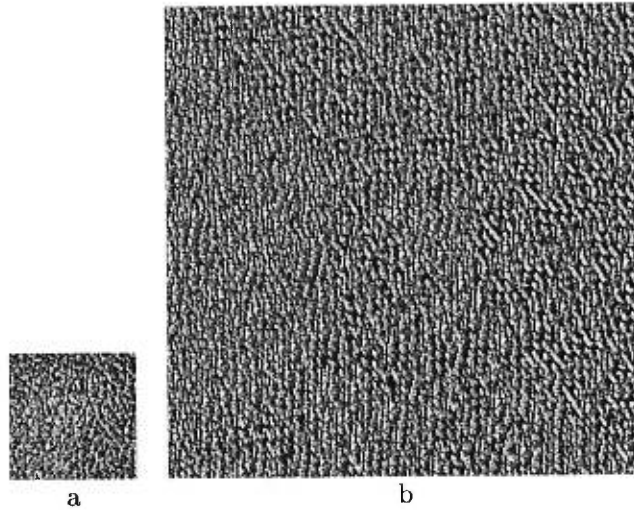


FIG. 4.5: (a) image d'entrée, (b) image synthétique

relie l'intensité de radiance (contenue dans la carte de radiance) à la hauteur des points sur la surface synthétique. L'utilisateur doit aussi entrer la couleur de la réflexion spéculaire. Ces informations pourraient être extraites automatiquement avec des données statistiques obtenues grâce à la carte de radiance. La Figure 4.6 donne des exemples de cartes de relief synthétisées.

Dans ce chapitre, nous avons proposé une méthode semi-automatique d'extraction de carte de relief à partir d'une carte de radiance d'une réflexion spéculaire réelle. Étant donnée la carte de radiance d'une réflexion spéculaire nous sommes capables de reproduire synthétiquement à l'aide du modèle de Phong [Pho75] et d'une carte de relief une réflexion spéculaire similaire à celle présente dans la carte de radiance.

Dans le Chapitre 5, nous présentons des exemples montrant l'application de cartes de relief à nos textures structurales.

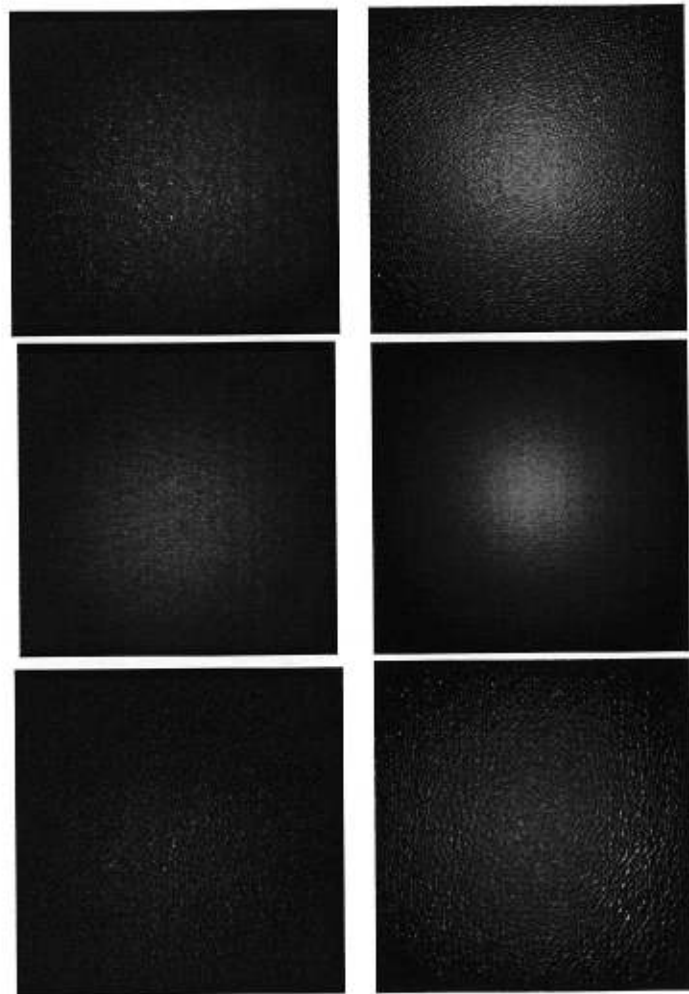


FIG. 4.6: Trois exemples de synthèse de cartes de relief. Image d'entrée à gauche, résultat à droite

## Chapitre 5

# Résultats et améliorations des modèles

*[...] an answer has more meaning to someone who has first asked a question.*

The Psychology of Learning Mathematics, *Richard R. Skemp*

Dans ce chapitre, nous présentons les résultats ainsi que quelques améliorations manuelles que nous pouvons apporter à nos modèles.

Modifier l'apparence d'une texture couleur demande beaucoup de travail de la part d'un artiste. Une fois les paramètres de texture extraits, nous pouvons plus facilement manuellement améliorer nos modèles car nous avons en main une définition structurelle de la texture. De plus, il est aussi possible d'utiliser une texture couleur en conjonction avec nos textures procédurales structurelles afin de créer certains effets.

Nous allons donner à travers ce chapitre quelques façons d'ajouter manuellement du réalisme à nos textures. Premièrement, nous montrerons comment altérer les normales de la surface pour créer l'illusion de micro-détails de surface, ensuite nous montrerons que nos textures procédurales peuvent être alignées entre des faces adjacentes, que l'on peut mélanger une texture couleur avec nos textures procédurales et enfin comment utiliser plusieurs textures structurelles pour en créer une autre plus complexe.

## 5.1 Application de cartes de micro-déplacement et de relief

Une carte de micro-déplacement ou une carte de relief peuvent être utilisées afin d'augmenter le réalisme de la texture générée. Par exemple, afin d'augmenter le réalisme d'un mur de briques, il est facile d'ajouter un léger bruit de surface sur les briques et de forcer le ciment à être plus enfoncé que les briques (comme c'est souvent le cas dans les murs de briques réels). La Figure 5.1 donne un exemple de l'utilisation d'une carte de relief et d'une carte de micro-déplacement sur un mur de briques. Ce mur est le même que celui montré au Chapitre 3 (Figure 3.5 p. 31) mais apparaît beaucoup plus réaliste à cause de ces ajouts. Notons cependant que ces cartes ne sont pas générées automatiquement par nos algorithmes et doivent donc être ajoutées manuellement par l'utilisateur.

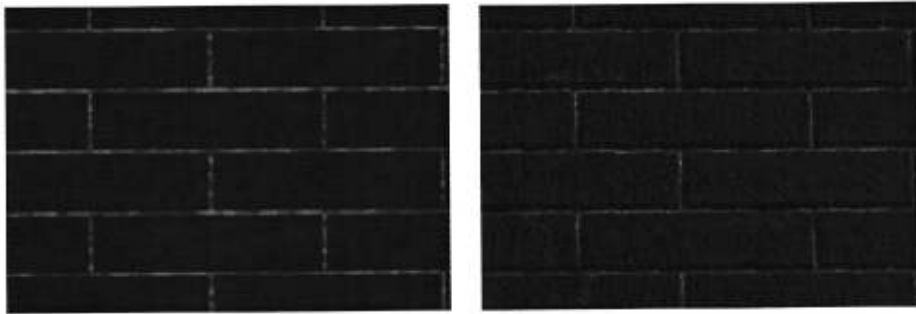


FIG. 5.1: Textures de briques synthétiques augmentées. À gauche une carte de relief est appliquée, à droite une carte de micro-déplacement est appliquée

Le prochain exemple montre comment une carte de relief permet de simuler la rugosité de tuiles de céramique. Dans cet exemple (Figure 5.2), de la turbulence perturbe les normales des tuiles de céramique mais ne touche pas au ciment entre les tuiles. Ceci permet d'avoir un reflet semblable à l'image originale et augmente grandement le réalisme de la texture générée par notre algorithme.

## 5.2 Alignement entre les textures de faces adjacentes

Un avantage important des textures procédurales est qu'il est possible d'ajouter des contraintes pour satisfaire la nature structurelle de la texture. Par exemple, lorsque



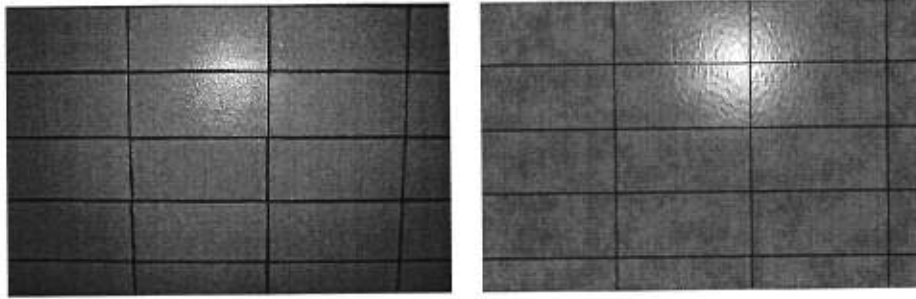


FIG. 5.2: Un vrai carrelage de céramique (à gauche) et la version synthétique de ce carrelage augmentée d'une carte de relief (à droite)

l'on applique une texture de briques sur deux murs adjacents on peut vouloir que les briques soient bien alignées entre les deux murs. La Figure 5.3 montre l'application de briques sur une tour en s'assurant que les rangées de briques s'alignent d'un mur à l'autre et que le décalage relatif entre les rangées de briques soit inversé entre les deux murs afin que la jonction soit naturelle entre ces derniers (on suppose ici que le décalage relatif entre deux rangées de briques correspond à l'épaisseur d'une brique). Une carte de micro-déplacement a aussi été ajoutée afin d'augmenter le réalisme des textures.

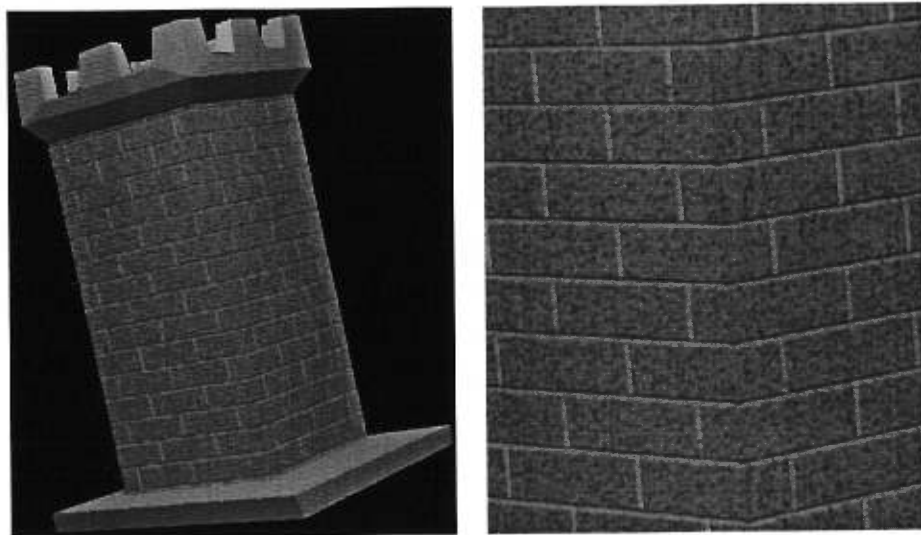


FIG. 5.3: Murs de briques sur une tour et vue rapprochée

### 5.3 Textures procédurales et textures couleur

Même si les textures procédurales sont un puissant outil afin de générer des textures structurelles, nombre de textures ne peuvent être modélisées par des procédures.

Quand nous rencontrons ce type de texture, l'utilisateur a la possibilité de choisir dans l'espace texture quelle partie doit être générée procéduralement et quelle partie doit être extraite directement des photographies comme une texture couleur (voir p. 6 pour plus d'explications quant aux différents types de textures). Afin de réaliser cette combinaison de textures, l'utilisateur peint un masque qui indique les régions à associer avec chaque type de texture. Le cadre de la salle de bain (Figures 5.9 et 5.10) ainsi que la grille métallique devant le foyer (Figure 5.12) sont des exemples de l'utilisation de cette méthode.

### 5.4 Textures structurelles multiples

Sur certains modèles, de multiples types de textures structurelles peuvent être appliquées sur la même surface tridimensionnelle. Les planchers de bois franc exploitent cette propriété en utilisant à la fois le bois et le carrelage rectangulaire dans leur génération.

La méthode de carrelage rectangulaire extrait la largeur bien démarquée dans la FFT à cause de la régularité (chaque planche est de même largeur) de chacune des planches et la technique de modélisation de bois remplit ces dernières. Étant donné que moins d'information est disponible dans une planche que dans un panneau de bois, le bois généré dans les planches peut parfois être moins précis. Par contre, nous avons trouvé que, à cause de la nature du bois, le résultat est satisfaisant même si des planches relativement étroites sont utilisées pour la synthèse. La Figure 5.4 montre une photographie d'un plancher de chêne que nous avons utilisé pour synthétiser les planchers de la Figure 5.5.

Quand les photographies de planchers ne donnent pas assez d'information sur la nature du bois, il est possible d'utiliser un panneau de ce même bois (dans une autre photographie) afin d'avoir plus d'information sur le bois à générer et appliquer ce dernier sur le plancher.



FIG. 5.4: Vrai plancher de chêne

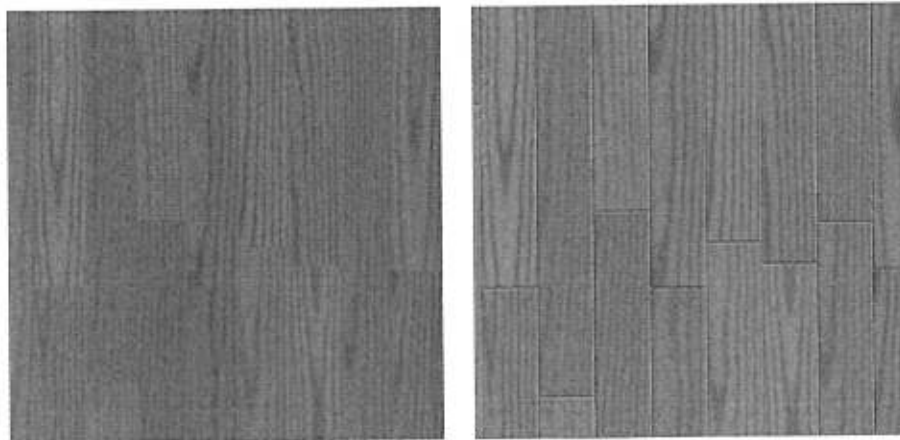


FIG. 5.5: Planchers synthétiques. Une carte de relief a été appliquée afin de courber les côtés des planches dans la figure de droite

## 5.5 Quelques exemples

Dans cette section, nous allons donner quelques exemples afin d'illustrer l'applicabilité de la méthode.

### 5.5.1 La salle de bain

Le premier exemple que nous présentons est celui d'une salle de bain modèle. La modélisation s'est faite à partir de 6 photographies (la Figure 5.6 représente une des 6 photographies utilisées) et de notre logiciel de reconstruction à partir de photographies [POF98].



FIG. 5.6: Une des photographies utilisées pour reconstruire la salle de bain

Le modèle tridimensionnel résultant est présenté à la Figure 5.7. Nous pouvons remarquer de nombreux problèmes reliés à la façon dont ont été extraites les textures comme par exemple des discontinuités sur les textures des murs car les objets à l'intérieur de la salle de bain n'ont pas été modélisés. Afin de corriger ces problèmes, nous utilisons notre méthode pour synthétiser de nouvelles textures. La première étape est de prendre des vues rapprochées des murs et du sol (voir Figure 5.8) afin d'avoir un bon niveau de détail pour l'analyse des textures structurelles à générer.

La Figure 5.9 présente le résultat obtenu suite à l'analyse et la synthèse des textures des murs et du sol de la salle de bain. Une carte de relief a été ajoutée par l'utilisateur afin d'augmenter le réalisme des textures. On peut remarquer que les problèmes présents



FIG. 5.7: La salle de bain reconstruite avec les textures projetées directement sur les murs

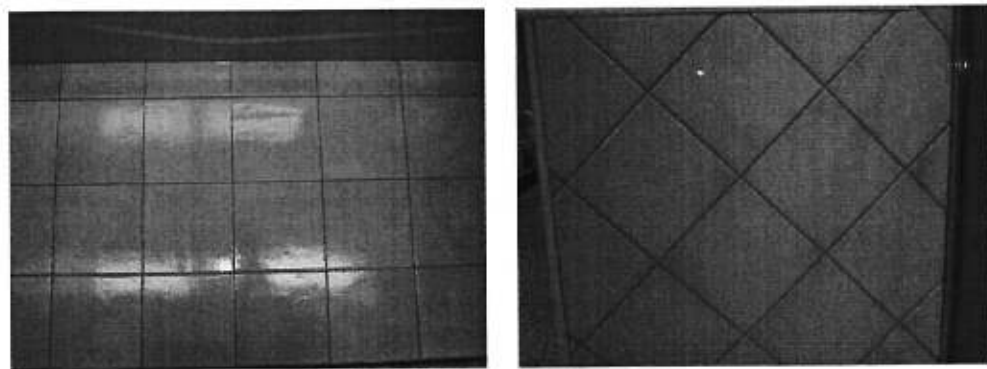


FIG. 5.8: Vue rapprochée du sol (à gauche) et d'un des murs (à droite)

dans la Figure 5.7 ont disparu, de plus la réflexion sur les murs est correctement calculée et les textures sur les murs s'alignent parfaitement entre les différentes faces. Le rendu de la scène se fait par lancer de rayons et une forme de *texture d'environnement* est utilisée hors du modèle pour simuler le reste de la pièce.



FIG. 5.9: Trois vues sur la salle de bain avec nos textures synthétiques

Afin de tester l'apparence de différentes tuiles nous pouvons facilement, comme l'illustre la Figure 5.10, prendre des échantillons d'autres tuiles ou même de bois et les apposer sur le plancher (ou sur les murs) de notre salle de bain modèle. Cette caractéristique de notre méthode peut être très utile en design ou redesign de pièces où l'utilisateur veut voir l'effet de certaines tuiles ou de bois dans un environnement donné.



FIG. 5.10: Substitution du plancher avec des textures extraites d'autres photographies

À cause des limitations de l'implantation courante de notre système de reconstruction à partir de photographies [POF98], les tuiles des Figures 5.9 et 5.10 apparaissent déformées (losanges plutôt que carrées). Ceci est dû au fait que les textures extraites par notre système de reconstruction à partir de photographies [POF98] sont distordues. Les algorithmes que nous proposons cependant dans le Chapitre 3 génèrent bien

des tuiles rectangulaires et une implantation de ces algorithmes sans les limitations de l'implantation courante de notre système de reconstruction à partir de photographies [POF98] ne causerait pas de distortion.

### 5.5.2 Le foyer

Le deuxième exemple que nous présentons est un foyer simple. Cet exemple illustre comment on peut, en ayant une seule photographie d'une scène contenant du bois et des briques, enlever des objets dans une scène (dans ce cas la tablette et les objets qui sont dessus) et ce sans avoir de trou dans les textures résultantes. La Figure 5.11 montre la photographie originale et la Figure 5.12 montre deux vues synthétiques du foyer et de la cheminée sans la tablette.



FIG. 5.11: Une photographie d'un foyer

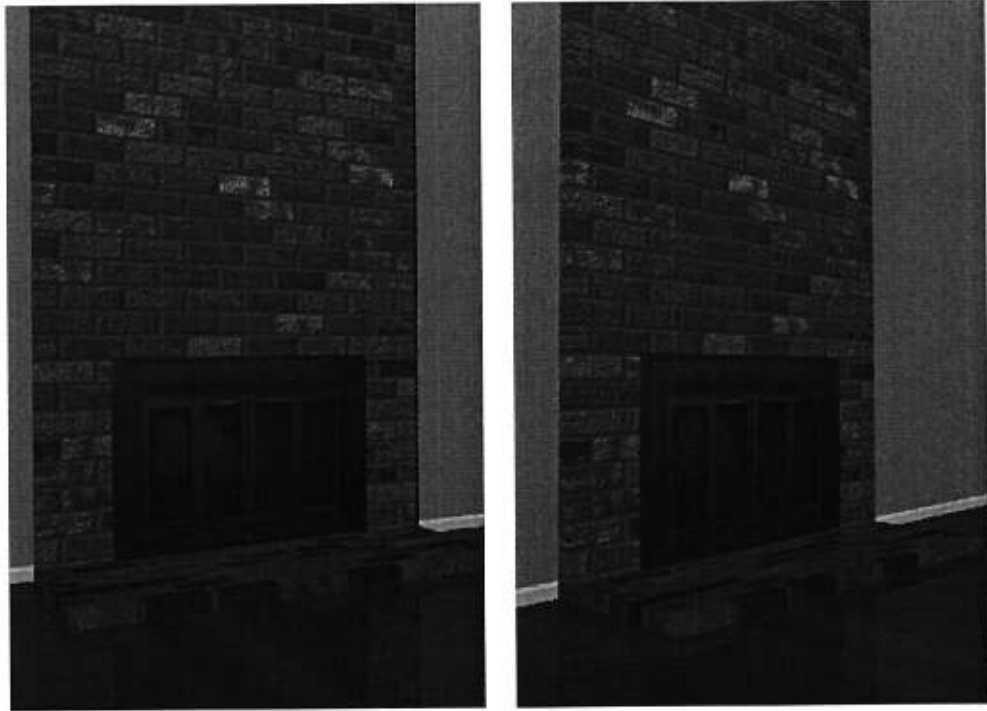


FIG. 5.12: Deux vues synthétiques du foyer

Dans ce chapitre, nous avons montré divers résultats obtenus grâce à notre méthode d'analyse et de synthèse de textures structurales. Rappelons que le but recherché de la méthode n'est pas nécessairement d'obtenir une texture qui est la même que la texture originale, mais qui y ressemble grandement.

Les modèles que nous avons proposés peuvent évidemment être améliorés afin d'obtenir des textures encore plus réalistes. Par exemple, le modèle de carrelage rectangulaire pourrait permettre de modifier la géométrie de chaque brique afin de simuler un vieux mur usé ou d'utiliser des tuiles non rectangulaires.



## Chapitre 6

# Conclusion

Les textures sont importantes en informatique graphique pour simuler la complexité à faible coût. Nous avons présenté une méthode d'analyse et de synthèse de deux grandes classes de textures structurelles : le carrelage rectangulaire et le bois. Nous avons choisi ces deux types de textures en raison de leur importance dans presque toutes les constructions humaines.

Notre méthode synthétise des textures procédurales de qualité qui règlent un bon nombre des problèmes reliés à l'extraction directe de textures à partir de photographies, à savoir :

1. Nos textures sont procédurales et ne sont donc pas limitées par la résolution de la photographie originale.
2. Nos textures n'ont aucun trou dû à l'occlusion car nous extrayons les paramètres de la texture plutôt que ses couleurs.
3. Nous pouvons assez facilement augmenter le réalisme de nos textures à l'aide d'une carte de relief ou d'une carte de micro-déplacement car nous avons une définition précise de chacun des éléments présents dans la texture.

Bien que notre méthode soit limitée à ces deux grandes classes de textures, nous pensons que les outils que nous avons introduits afin d'extraire les paramètres de nos textures peuvent être réutilisés pour l'extraction des paramètres d'autres types de textures. De plus, l'implantation d'un nouveau type de texture structurelle ne devrait pas prendre beaucoup de temps étant donné que tous les outils fournis pour segmenter une image en un masque binaire peuvent être réutilisés tel quel et que les structures utilisées

pour la synthèse d'images et l'interface sont déjà implantées.

## 6.1 Améliorations et extensions

### 6.1.1 Textures structurelles hiérarchiques

Pour le moment, nous sommes limités à nos deux grandes classes de textures. Il serait cependant possible de construire une hiérarchie de textures afin de synthétiser des textures structurelles plus complexes. Par exemple, plutôt que d'utiliser simplement l'algorithme de Heeger et Bergen [HB95] pour remplir l'intérieur de nos tuiles, on pourrait créer des nouveaux types de textures structurelles et utiliser ces dernières pour remplir l'intérieur des carreaux rectangulaires par une mosaïque de petites tuiles. Nous aurions à ce moment une texture à plusieurs niveaux. Au premier niveau, un carrelage rectangulaire et dans chaque case du carrelage (au deuxième niveau) une mosaïque.

### 6.1.2 Extraction des zones d'ombre

En utilisant notre logiciel de reconstruction à partir de photographies [POF98], il serait possible de détecter la position des lumières (en les prenant en photographie en relation avec d'autres objets de la scène) dans la scène ainsi que celle des objets. À ce moment, nous pourrions savoir quelles sont les portions de textures qui sont dans l'ombre sur la photographie et quelles portions sont illuminées. Ces informations pourraient par exemple nous guider dans le choix des tuiles à utiliser pour la synthèse plutôt que de demander à l'utilisateur de nous les indiquer.

Cette méthode nous permettrait d'éviter les zones d'ombres lors du choix des tuiles dans l'image originale. Nous pourrions aussi penser à un algorithme de détection des reflets basé sur la position relative entre la caméra, les lumières et les objets de la scène et ainsi éviter aussi ce type d'artefact lors de la sélection de nos tuiles.

### 6.1.3 Extraction de cartes de relief sur des objets texturés

La méthode d'extraction de cartes de relief proposée au Chapitre 4 ne fonctionne que sur des objets planaires de couleur uniforme (les changements de couleur ne sont dus

qu'au micro-relief de la surface). À l'aide d'un algorithme à deux passes nous pourrions extraire le relief d'objets texturés et ainsi utiliser cette technique pour extraire la carte de relief de nos textures plutôt que de demander à l'utilisateur de la stipuler manuellement. Dans la première passe de l'algorithme une image de la surface à étudier serait prise sous un éclairage diffus (non directionnel) sur lequel on ne voit que la texture. Dans la deuxième passe on prend une photographie de la surface éclairée par une lumière ponctuelle (comme cela est le cas dans le Chapitre 4). Il ne nous reste alors qu'à soustraire la première image de la deuxième afin d'enlever la texture de la surface et de ne conserver ainsi que les changements de couleurs dus au micro-relief de la surface.

## Annexe A

# Glossaire

Ce glossaire donne la définition utilisée dans le présent document pour certains termes. Comme l'utilisation du français en infographie peut parfois porter à confusion, les termes traduits ou empruntés de l'anglais se retrouvent ici.

**Aliassage** *aliasing*

**Apposition de textures** *texture mapping*

**Bois d'été** *Late wood* Le bois d'été est formé à la fin du cycle de croissance et il en résulte que ce dernier est formé de plus petites cellules plus denses. Le bois d'été est habituellement de couleur plus foncée que le bois de printemps.

**Bois de printemps** *Early wood* Le bois de printemps est formé au début du cycle de croissance annuel alors que beaucoup de sève est disponible et que la croissance est rapide. Les cellules formées sont plus grandes. Le bois de printemps est habituellement de couleur plus pâle que le bois d'été.

**BRDF** *Bidirectional reflectance distribution function* Fonction qui décrit comment une surface interagit avec la lumière. Plus précisément, il s'agit de la proportion de lumière réfléchiée dans une direction étant donnée la direction de lumière incidente.

**Carte de micro-déplacement** *Displacement map* Une carte de micro-déplacement est une façon d'ajouter un micro-relief à une surface. La surface est divisée en micro-polygones et la hauteur des polygones est perturbée en suivant une carte de hauteur (*height field* en anglais).

**Carte de relief** *Bump map* Une carte de relief est la même chose qu'une carte de

micro-déplacement à la différence que seules les normales sont perturbées par la carte de hauteur.

**Informatique graphique** *Computer Graphics*

**Ligne de balayage** *Scanline* Ligne de pixels consécutifs situés sur une ligne de l'image, soit verticale, soit horizontale.

**Matériel graphique** *graphics hardware*

**Pyramide navigable** *Steerable Pyramid* Transformation qui décompose une image en pyramide de fonctions de base pondérées qui ne sont pas radialement symétriques (voir p. 12 Section 2.3.1 pour plus de détails).

**Texture** Altération des propriétés (couleurs, normales, etc.) d'un polygone en utilisant une image ou un algorithme pour ce faire.

**Texture basée sur une carte d'information** *raster texture* Altération des paramètres d'une surface en utilisant une carte d'information comme référence et une paramétrisation donnée. Ces informations peuvent être des couleurs, orientations de normales, micro-déplacements de la surface, propriétés de réflexion, transparence, etc.

**Texture couleur** Altération des couleurs d'une surface en utilisant une image comme référence et une paramétrisation donnée (sous ensemble des textures basées sur une carte d'information).

**Texture d'environnement** *Environnement map* Texture généralement apposée sur un hémisphère (ou une sphère) englobant la scène et qui simule l'environnement. On s'en sert beaucoup en informatique graphique pour simuler les réflexions en temps réel.

**Texture procédurale** Altération des couleurs d'une surface en utilisant un algorithme conçu par l'artiste.

**Texture stochastique** Texture générée à partir d'un processus aléatoire et donc qui n'est pas composée de motifs répétitifs simples.

**Texture structurelle** Texture qui est composée de motifs répétitifs avec une sémantique plus ou moins précise dépendant de la texture.

# Bibliographie

- [BA83] P. J. Burt et E. H. Adelson. « The Laplacian pyramid as a compact image code ». *IEEE Transactions on Communications*, volume COM-31,4, pages 532–540, avril 1983.
- [BG89] C. Bennis et A. Gagalowicz. « 2D macroscopic texture synthesis ». *Computer Graphics Forum*, volume 8, numéro 4, pages 291–300, décembre 1989.
- [Bli78] J. F. Blinn. « Simulation of wrinkled surfaces ». Dans *SIGGRAPH 1978 Conference Proceedings*, volume 12, pages 286–292, août 1978.
- [Bli89] J. F. Blinn. « Jim Blinn’s corner : return of the jaggy ». *IEEE Computer Graphics and Applications*, volume 9, numéro 2, pages 82–89, mars 1989.
- [BM93] B. G. Becker et N. L. Max. « Smooth transitions between bump rendering algorithms ». Dans *SIGGRAPH 1993 Conference Proceedings*, volume 27, pages 183–190, août 1993.
- [BN76] J. F. Blinn et M. E. Newell. « Texture and reflection in computer generated images ». *Communications of the ACM*, volume 19, pages 542–546, 1976.
- [Buc98] J. W. Buchanan. « Simulating wood using a voxel approach ». Dans *Eurographics 1998*, volume 17, pages C105–C112, 1998.
- [Cat74] E. E. Catmull. *A subdivision algorithm for computer display of curved surfaces*. Ph.D. thesis, University of Utah, décembre 1974.
- [CJ83] G. R. Cross et A. K. Jain. « Markov random field texture models ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-5, numéro 1, pages 25–39, janvier 1983.

- [CK85] R. Chellappa et R. L. Kashyap. « Texture synthesis using 2-D noncausal autoregressive models ». *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 33, pages 194–203, 1985.
- [Coo84] R. L. Cook. « Shade trees ». Dans *SIGGRAPH 1984 Conference Proceedings*, volume 18, pages 223–231, juillet 1984.
- [De 97] J. S. De Bonet. « Multiresolution sampling procedure for analysis and synthesis of texture images ». Dans *SIGGRAPH 1997 Conference Proceedings*, Annual Conference Series, pages 361–368, août 1997.
- [DGF98] J.M. Dischler, D. Ghazanfarpour et R. Freydier. « Anisotropic solid texture synthesis using orthogonal 2D views ». Dans *Eurographics 1998*, volume 17, pages C87–C95, 1998.
- [DM97] P.E. Debevec et J. Malik. « Recovering high dynamic range radiance maps from photographs ». Dans *SIGGRAPH 1997 Conference Proceedings*, numéro 3 in Annual Conference Series, pages 369–378, août 1997.
- [DTM96] P. E. Debevec, C. J. Taylor et J. Malik. « Modeling and rendering architecture from photographs : a hybrid geometry- and image-based approach ». Dans *SIGGRAPH 1996 Conference Proceedings*, Annual Conference Series, pages 11–20, août 1996.
- [DYB98] P. E. Debevec, Y. Y. et G. D. Borshukov. « Efficient view-dependent image-based rendering with projective texture-mapping ». Dans *Rendering Techniques 1998*, Eurographics, pages 105–116, 1998.
- [EL99] A.A. Efros et T. K. Leung. « Texture synthesis by non-parametric sampling ». Dans *IEEE International Conference on Computer Vision (ICCV'99)*, pages 13–21, septembre 1999.
- [EM97] T. El-Maraghi. « An Implementation of Heeger and Bergen's texture analysis/synthesis algorithm with source code ». *University of Toronto Technical Report*, [www.cs.utoronto.ca/~tem/2522/texture.html](http://www.cs.utoronto.ca/~tem/2522/texture.html), septembre 1997.
- [EMP<sup>+</sup>94] D. Ebert, K. Musgrave, D. Peachey, K. Perlin et S. Worley. *Texturing and modeling : a procedural approach*. Academic Press, octobre 1994.

- [FFC82] A. Fournier, D. Fussell et L. Carpenter. « Computer rendering of stochastic models ». *Communications of the ACM*, volume 25, numéro 6, pages 371–384, juin 1982.
- [FLCB95] K. Fleischer, D. Laidlaw, B. Currin et A. Barr. « Cellular texture generation ». Dans *SIGGRAPH 1995 Conference Proceedings*, volume 14, pages 239–248, August 1995.
- [Fou92] A. Fournier. « Filtering normal maps and creating multiple surfaces ». Technical Report Imager 92/1, Imager, 1992.
- [Gag87] A. Gagalowicz. « Texture modelling applications ». *The Visual Computer*, volume 3, numéro 4, pages 186–200, décembre 1987.
- [GM85] A. Gagalowicz et S. De Ma. « Sequential synthesis of natural textures ». *Computer Vision, Graphics, and Image Processing*, volume 30, numéro 3, pages 289–315, juin 1985.
- [HB95] D. J. Heeger et J. R. Bergen. « Pyramid-based texture analysis/synthesis ». Dans *SIGGRAPH 1995 Conference Proceedings*, Annual Conference Series, pages 229–238, août 1995.
- [Hec86] P. S. Heckbert. « Survey of statistics mapping ». *IEEE Computer Graphics and Applications*, volume 6, numéro 11, pages 56–67, 1986.
- [HLM96] P. Havaldar, M.S. Lee et G. Medioni. « View synthesis from unregistered 2-D images ». Dans *Graphics Interface 1996*, pages 61–69, mai 1996.
- [JFP93] A.Z. Meiri J.M. Francos et B. Porat. « A unified texture model based on a 2-D wold-like decomposition ». *IEEE Trans. Signal Processing*, volume 41, numéro 8, pages 2665–2678, août 1993.
- [KMR72] R. Karp, R. Miller et A. Rosenberg. « Rapid identification of repeated patterns in strings, trees, and arrays ». Dans *Proc. of the ACM Symposium on the Theory of Computing*, pages 125–136, 1972.
- [Lew84] J.P. Lewis. « Texture synthesis for digital painting ». Dans *SIGGRAPH 1984 Conference Proceedings*, volume 18, pages 245–252, juillet 1984.
- [Lew87] J. P. Lewis. « Generalized stochastic subdivision ». *ACM Transactions on Graphics*, volume 6, pages 167–190, juillet 1987.



- [LFD<sup>+</sup>99] C. Loscos, M.-C. Frasson, G. Drettakis, B. Walter, X. Granier et P. Poulin. « Interactive virtual relighting and remodeling of real scenes ». Dans *Tenth Eurographics Workshop on Rendering*, pages 329–340, juin 1999.
- [Pea85] D. R. Peachey. « Solid texturing of complex surfaces ». Dans *SIGGRAPH 1985 Conference Proceedings*, volume 19, pages 279–286, juillet 1985.
- [Pen88] A. Pentland. « On the extraction of shape information from shading ». Dans *National Conference on Artificial Intelligence*, pages 826–830, 1988.
- [Per85] K. Perlin. « An image synthesizer ». Dans *SIGGRAPH 1985 Conference Proceedings*, volume 19, pages 287–296, juillet 1985.
- [PH89] K. Perlin et E. M. Hoffert. « Hypertexture ». Dans *SIGGRAPH 1989 Conference Proceedings*, volume 23, pages 253–262, juillet 1989.
- [Pho75] B-T. Phong. « Illumination for computer generated pictures ». *CACM June 1975*, volume 18, numéro 6, pages 311–317, 1975.
- [POF98] P. Poulin, M. Ouimet et M.C. Frasson. « Interactively Modeling with Photogrammetry ». Dans *Ninth Eurographics Workshop on Rendering*, pages 93–104, Vienna, Austria, juin 1998.
- [PP93] K. Popat et R. W. Picard. « Novel cluster-based probability model for texture synthesis, classification, and compression ». Rapport technique 234, MIT Media Lab Vismod, 1993.
- [RTG97] H. Rushmeier, G. Taubin et A. Guézic. « Applying shape from lighting variation to bump map capture ». Dans *Eurographics Rendering Workshop 1997*, pages 35–44, juin 1997.
- [SFAH92] E. P. Simoncelli, W. T. Freeman, E. H. Adelson et D. J. Heeger. « Shiftable multi-scale transforms ». *IEEE Trans. Information Theory*, volume 38, numéro 2, pages 587–607, mars 1992.
- [SWI97] Y. Sato, M. D. Wheeler et K. Ikeuchi. « Object shape and reflectance modeling from observation ». Dans *SIGGRAPH 1997 Conference Proceedings*, pages 379–388, 1997.

- [TS67] K. E. Torrance et E. M. Sparrow. « Theory for off-specular reflection from roughened surfaces ». *Journal of Optical Society of America*, volume 57, numéro 9, 1967.
- [Tur91] G. Turk. « Generating textures for arbitrary surfaces using reaction-diffusion ». Dans *SIGGRAPH 1991 Conference Proceedings*, volume 25, pages 289–298, juillet 1991.
- [Umb98] S. E. Umbaugh. *Computer vision and image processing*. Prentice-Hall Inc., 1998.
- [War92] G. J. Ward. « Measuring and modeling anisotropic reflection ». Dans *SIGGRAPH 1992 Conference Proceedings*, volume 26, pages 265–272, juillet 1992.
- [Wei90] D. Weinshall. « The shape of shading ». Technical Report AIM-1264, Massachusetts Institute of Technology, octobre 1990.
- [Wei00] L. Wei. « An implementation of Alyosha Efros texture synthesis algorithm ». Rapport technique, janvier 2000.
- [Whi85] T. Whitted. « The hacker’s guide to making pretty pictures ». Dans *SIGGRAPH 1985 Image Rendering Tricks seminar notes*. Addison Wesley, juillet 1985.
- [Wil83] L. Williams. « Pyramidal parametrics ». Dans *SIGGRAPH 1983 Conference Proceedings*, volume 17, pages 1–11, juillet 1983.
- [WK91] A. Witkin et M. Kass. « Reaction-diffusion textures ». Dans *SIGGRAPH 1991 Conference Proceedings*, volume 25, pages 299–308, juillet 1991.
- [WL00] L.Y. Wei et M. Levoy. « Fast texture synthesis using tree-structured vector quantization ». Dans *SIGGRAPH 2000 Conference Proceedings*, volume 34, juillet 2000.
- [YDMH99] Y. Yu, P. Debevec, J. Malik et T. Hawkins. « Inverse global illumination : recovering reflectance models of real scenes from photographs ». Dans *SIGGRAPH 1999 Conference Proceedings*, volume 33, pages 215–224, août 1999.

- [ZZ88] W. Zhang et Y. Zhou. « An improved method for shape from shading ». Dans *Ninth International Conference on Pattern Recognition*, pages 1106–1108, 1988.