



Université de Montréal

# Extraction et Unification de Textures en Provenance d'Images

par

Mathieu Ouimet

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

Septembre 1998

© Mathieu Ouimet, 1998

Université de Montréal  
Faculté des études supérieures

Ce mémoire de maîtrise intitulé

Extraction et Unification de Textures  
en Provenance d'Images

présenté par  
Mathieu Ouimet

a été évalué par un jury composé des personnes suivantes :

Président : Neil Stewart

Directeur de recherche : Pierre Poulin

Membre : Jean Meunier

# Sommaire

La plupart des images synthétiques qu'on peut voir en infographie représentent le rendu d'objets géométriques complexes qu'il a fallu modéliser. La complexité des modèles, d'où le réalisme des images est directement issu, ne concerne pas seulement la précision et la diversité de la représentation géométrique mais aussi la description texturale des modèles. L'apposition de textures sur la surface d'un modèle est une technique incontournable en infographie pour augmenter rapidement et simplement le réalisme des modèles et par conséquent les images synthétiques de ces derniers.

Il y a quelques années déjà, plusieurs techniques ont appuyé le domaine de la modélisation de solides en présentant le concept de reconstruction d'objets à partir d'images et plus particulièrement à partir de photographies. Notre recherche se situe au niveau de la reconstruction des textures des modèles contenus dans les images. Un modèle géométrique représenté dans plusieurs images est reconstruit et sa projection nous permet d'extraire les couleurs constituant sa texture dans chaque image. En plus de l'échantillonnage des textures dans les différentes images, notre recherche concerne le problème d'unification des textures d'une même facette polygonale puisqu'on accepte utilement une texture unique par facette d'un objet et que cette dernière est contenue dans plusieurs images provenant de points de vue différents. La variété des images pouvant être utilisées par la reconstruction nous amène à considérer plusieurs facteurs tels la résolution des images et leur niveau de précision, les différentes occlusions à la fois avec les bordures des images et avec les primitives contenues dans celles-ci, les différences d'illumination dues à la direction des points de vue et aux changements de radiance ou de position des sources lumineuses, la correction des couleurs et les diverses corruptions des informations causées par les procédés photographiques ou infographiques. Notre technique vise, en partie, l'utilisation optimale des informations contenues dans les images afin de générer des textures représentant le plus fidèlement possible les couleurs d'origine du modèle.

## Mots-clés :

Infographie, texture, modélisation, réalisme, reconstruction, image.

# Table des matières

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>Remerciements</b>                                                | <b>vi</b> |
| <b>1 Introduction</b>                                               | <b>1</b>  |
| 1.1 Importance des modèles géométriques en infographie . . . . .    | 1         |
| 1.2 Les textures comme solution au problème de complexité . . . . . | 2         |
| 1.3 La modélisation et l'extraction des textures . . . . .          | 3         |
| 1.4 Présentation du travail . . . . .                               | 4         |
| <b>2 Modélisation à partir d'images</b>                             | <b>7</b>  |
| 2.1 Interpolation de vues et rendu à base d'images . . . . .        | 8         |
| 2.1.1 Techniques de rendu par interpolation d'images . . . . .      | 9         |
| 2.1.2 Techniques de rendu par champs de lumière . . . . .           | 10        |
| 2.2 Projection de textures . . . . .                                | 12        |
| <b>3 Reconstruction géométrique</b>                                 | <b>16</b> |
| 3.1 Mise en situation . . . . .                                     | 17        |
| 3.2 Méthodes épipolaires . . . . .                                  | 20        |
| 3.2.1 Recherche sur les lignes épipolaires . . . . .                | 20        |
| 3.2.2 Problèmes de la technique épipolaire . . . . .                | 23        |
| 3.3 Méthodes par contraintes . . . . .                              | 25        |
| 3.4 Matrices de projection . . . . .                                | 27        |
| <b>4 Reconstruction des textures</b>                                | <b>30</b> |
| 4.1 Théorie des textures . . . . .                                  | 32        |
| 4.2 Extraction des textures . . . . .                               | 35        |
| 4.2.1 Échantillonnage et filtrage . . . . .                         | 36        |

|          |                                                     |           |
|----------|-----------------------------------------------------|-----------|
| 4.2.2    | Construction des texels . . . . .                   | 39        |
| 4.3      | Unification des textures . . . . .                  | 44        |
| 4.3.1    | Transformations géométriques des textures . . . . . | 44        |
| 4.3.2    | Différences de couleurs . . . . .                   | 47        |
| <b>5</b> | <b>Conclusion</b>                                   | <b>60</b> |
| <b>A</b> | <b>Description du logiciel</b>                      | <b>64</b> |
| A.1      | Système de reconstruction géométrique . . . . .     | 65        |
| A.2      | Système de reconstruction des textures . . . . .    | 67        |
| A.2.1    | Structures de données . . . . .                     | 68        |
| A.2.2    | Extraction des données . . . . .                    | 69        |
| A.2.3    | Unification . . . . .                               | 71        |
| A.2.4    | Remplissage . . . . .                               | 73        |
| A.3      | Système général . . . . .                           | 75        |
| A.3.1    | Analyse et design . . . . .                         | 76        |
| A.3.2    | Programmation . . . . .                             | 79        |
| <b>B</b> | <b>Méthode des moindres carrés</b>                  | <b>81</b> |
| <b>C</b> | <b>Calcul de la caméra d'une image</b>              | <b>83</b> |
| C.1      | Calcul de la position d'une caméra . . . . .        | 83        |
|          | <b>Bibliographie</b>                                | <b>87</b> |

# Table des figures

|      |                                                                                    |    |
|------|------------------------------------------------------------------------------------|----|
| 2.1  | Environnements de projection . . . . .                                             | 10 |
| 2.2  | Paramétrisation par deux plans . . . . .                                           | 11 |
| 3.1  | Géométrie épipolaire . . . . .                                                     | 21 |
| 3.2  | Étapes de reconstruction géométrique . . . . .                                     | 28 |
| 4.1  | Texture procédurale . . . . .                                                      | 31 |
| 4.2  | Textures provenant de photographies . . . . .                                      | 33 |
| 4.3  | <i>Height field</i> . . . . .                                                      | 34 |
| 4.4  | Échantillonnage sans interpolation et avec interpolation bilinéaire . . . .        | 37 |
| 4.5  | Étapes d'échantillonnage d'une couleur dans une image . . . . .                    | 38 |
| 4.6  | Échantillonnage et filtrage en $3D$ et en $2D$ . . . . .                           | 39 |
| 4.7  | Filtre delta . . . . .                                                             | 39 |
| 4.8  | Calcul des occlusions par lancer de rayons . . . . .                               | 42 |
| 4.9  | <i>Résultats des différences entre une texture alignée et non alignée.</i> . . . . | 45 |
| 4.10 | L'espace de couleurs $RGB$ . . . . .                                               | 46 |
| 4.11 | Réflexions spéculaires sur une surface diélectrique (plastique) . . . . .          | 48 |
| 4.12 | Histogramme des couleurs et identification des réflexions spéculaires . .          | 49 |
| 4.13 | Schéma de réflexion spéculaire . . . . .                                           | 50 |
| 4.14 | Image de disparités . . . . .                                                      | 51 |
| 4.15 | Unification des texels avec écarts types et moyennes . . . . .                     | 53 |
| 4.16 | Configuration spatiale d'une caméra et d'un échantillon . . . . .                  | 57 |
| A.1  | Schéma de distribution des classes du système . . . . .                            | 66 |
| A.2  | Légende de la figure A.1 . . . . .                                                 | 67 |
| A.3  | WsHyperTexMap . . . . .                                                            | 68 |

|     |                                                                            |    |
|-----|----------------------------------------------------------------------------|----|
| A.4 | WsHyperTexel . . . . .                                                     | 69 |
| A.5 | Les voisins d'un texel . . . . .                                           | 74 |
| A.6 | Zones d'occlusions temporaires et permanentes . . . . .                    | 75 |
| A.7 | Nouvelle architecture du processus de reconstruction géométrique . . . . . | 77 |
| C.1 | Ligne dans une image formant un plan avec la caméra . . . . .              | 84 |
| C.2 | Deux plans définis à un point dans une image . . . . .                     | 84 |
| C.3 | Étapes de calcul de la direction de vue . . . . .                          | 85 |
| C.4 | Point de vue orthographique . . . . .                                      | 86 |

À Suzanne, Paul, Frédéric et Sébastien. Voilà le produit de toutes ces heures de travail hermétique, qui ont causé votre questionnement acharné pendant ces deux dernières années.

# Remerciements

L'acte de foi est à la base de tous les grands accomplissements. La transcendance de ces derniers dépend de cet élan primordial de confiance qui observe à travers l'avenir et qui nous indique la marche à suivre. Il n'est fondé sur aucun principe formel mais il respire encore plus la vérité. Je ne peux me prononcer sur la valeur de mon travail durant ces deux dernières années mais chose certaine, il reflète pâlement tout ce que j'ai acquis. Je serai toujours redevable à ceux qui m'ont offert une part de leur vision, qui ont partagé leur liberté de penser afin de me rendre moins aveugle. M. Pierre Poulin fait partie de ces gens charitables, qui sont capables d'actes de foi, qui m'ont donné une chance concrète de prouver mon potentiel et qui m'ont guidé vers le dépassement de mes capacités. J'espère vivre assez longtemps pour remercier M. Poulin et ma famille, en prouvant que leurs espoirs en mes aspirations et en mes rêves ne sont pas illusoire et en méritant la confiance qu'ils eurent envers moi. Le résultat véritable de toutes ces années d'étude réside dans l'infime clairvoyance que j'ai accumulée et dans la propension à créer qu'elle me procure. Voilà bien le plus grand des accomplissements.

# Chapitre 1

## Introduction

### 1.1 Importance des modèles géométriques en infographie

L’infographie est un domaine de l’informatique qui concerne la représentation visuelle d’objets formalisés dans un médium ordiné. Dans la majorité des cas, bien qu’on soit en présence d’entités à représentations autant abstraites que concrètes, l’infographie s’est surtout concentrée sur le rendu d’objets à représentations concrètes, c’est-à-dire des objets possiblement tangibles dans la réalité. Le formalisme utilisé pour décrire ces objets dans un système informatisé tient surtout de la géométrie et par conséquent, se rapproche souvent de la manière ostentatoire et visuelle que nous avons pour représenter mentalement les objets physiques.

Une forme de représentation d’objets intuitive, fort utilisée en infographie, est la représentation frontalière. Elle consiste en la description des objets par leur surface extérieure, c’est-à-dire les facettes polygonales ou courbes qui composent l’enveloppe géométrique de l’objet. Un des problèmes majeurs concernant le rendu des scènes composées de tels objets réside dans la complexité de ces derniers. Le réalisme d’un rendu dépend directement de la complexité des modèles qui sont impliqués et donc du nombre de facettes élémentaires qui sont utilisées pour composer la surface des objets. La création de modèles précis et complexes, requis par les méthodes de rendu réaliste, demande un effort humain considérable. Entre autres tâches laborieuses, l’usager doit spécifier de manière plus ou moins efficace les multiples facettes d’un modèle et les diverses relations entre elles. Les interfaces graphiques étant devenues plus intuitives et rendant le travail plus efficace, il n’en demeure pas moins que la modélisation subsiste en

tant que le plus grand tribut pour l'utilisateur dans le processus de synthèse d'images et revêt beaucoup d'importance en terme de réalisme. Un domaine de recherche en particulier, la modélisation de solides [Man88], tente de simplifier et d'accélérer le processus de construction des modèles synthétiques.

## 1.2 Les textures comme solution au problème de complexité

La complexité des objets qu'il est possible de modéliser ne s'approche en rien de celle des modèles réels, même ceux comportant très peu de facettes frontalières. Ceci vient du fait que les surfaces réelles ont non seulement une situation spatiale mais aussi une situation d'interaction avec la lumière, d'où les multiples propriétés découlant des nombreux modèles de réflexion et d'illumination en infographie, et une diversité très importante au niveau des motifs et textures qui décuplent la complexité d'une facette. Pour atteindre un réalisme visuel impératif en infographie, on ne peut représenter un polygone seulement par ses sommets, il faut véritablement définir ce qui se passe entre ces sommets, sur la surface elle-même. Une bonne partie de la complexité des modèles en infographie est prise en charge par les textures, ces "images" paramétrisées et apposées sur les facettes dans le but de représenter, à une résolution finie, les diverses propriétés de chaque élément de surface. Il est évident que chaque élément de surface infinitésimal d'un polygone ne peut être caractérisé individuellement. Toutefois, une texture est un excellent moyen de caractériser les comportements divers d'une surface à une certaine résolution.

Pour la majorité des techniques de rendu, déterminer la visibilité représente une tâche complexe et très coûteuse en mémoire et en temps. Le problème de visibilité est directement relié à la complexité des modèles utilisés. Cette affliction qu'est le calcul de visibilité pour les scènes très bien détaillées et qui comportent plusieurs facettes n'est toutefois pas ressentie par l'apposition des textures sur ces dernières. Ceci constitue, bien entendu, un avantage marqué des textures qui augmentent significativement la complexité des facettes sans imposer une hausse de travail pour la détermination de la visibilité et pour le rendu sous-jacent en général.

### 1.3 La modélisation et l'extraction des textures

Bien qu'on soit en mesure de représenter plus facilement et en détails le comportement des surfaces avec des textures au lieu de seulement avoir une représentation géométrique, il reste que la création des textures pose d'autres problèmes que la création des modèles uniquement géométriques. La modélisation d'une toile rectangulaire est géométriquement simple. L'utilisateur d'un système de modélisation de solides n'a qu'à définir quatre sommets correspondant aux quatre coins de la toile. Par contre, si on veut modéliser ce qu'il y a sur la toile, une peinture par exemple, alors l'utilisateur devra entrer la bonne couleur au bon endroit sur le rectangle, ce qui devrait habituellement exiger beaucoup plus d'efforts qu'une simple modélisation géométrique. Il incombe à l'utilisateur d'entrer et de manipuler beaucoup plus d'informations, donc ceci requiert nécessairement plus de temps. De plus, dans ce dernier cas, l'utilisateur n'a qu'à spécifier les couleurs de la surface à une certaine résolution ce qui est beaucoup moins exigeant que de tenter de modéliser géométriquement les différents traits et variations de couleurs qui se retrouvent dans la texture. Les textures offrent ainsi un bon compromis entre la modélisation trop simple d'une toile rectangulaire de couleur uniforme à une toile trop complexe où toutes les lignes, les traits de différentes couleurs sont modélisés géométriquement. Malgré tout, en intégrant les textures dans la description des modèles, on a non seulement augmenté la complexité de ces derniers ce qui nous donne des rendus beaucoup plus réalistes, mais on a aussi complexifié considérablement la tâche de modélisation. Pour chaque facette modélisée géométriquement, il faut maintenant lui attribuer une texture qui peut provenir d'une photographie par exemple, ou qui peut être synthétisée automatiquement ou créée par un artiste.

Depuis déjà un certain nombre d'années, les grands problèmes de la création de scènes complexes pour le rendu en infographie ont été cernés et simplifiés par de nombreuses techniques rassemblées sous la bannière de la reconstruction à partir d'images et plus particulièrement, à partir de photographies. D'une part, la modélisation géométrique est partiellement automatisée ou du moins simplifiée par l'utilisation d'images qui constituent des plans, des cartes sur lesquelles l'utilisateur a la possibilité de construire directement les modèles au lieu d'avoir à les manipuler abstraitement dans un espace euclidien plus ou moins intuitif. D'autre part, les images sont des traces photoniques (photographies) ou, plus généralement, des projections d'une scène prises

à certains points de vue. Par conséquent, toutes les informations de couleur concernant la surface d'une facette se retrouvent plus ou moins transformées et modifiées dans une image. De cette façon, en plus d'une aide pour la modélisation géométrique, les images d'une scène semblent nous offrir presque gratuitement les textures relatives à chaque facette observée dans la scène. La modélisation par images englobe bien cette dualité qui existe dans la définition des modèles, définition géométrique et définition texturale. En ce sens, les photographies peuvent être considérées au même niveau que les images de rendu puisqu'elles constituent des projections bidimensionnelles de scènes tridimensionnelles.

Bien que les contributions des techniques de reconstruction à partir d'images soient indéniables au niveau de la modélisation géométrique, beaucoup d'informations contenues dans les images sont typiquement ignorées. Ce sont ces informations un peu négligées qui nous intéressent dans ce travail qui concerne l'extraction et l'unification des textures en provenance d'images.

## 1.4 Présentation du travail

Les techniques de reconstruction à partir d'images sont normalement formées de deux étapes importantes : la reconstruction du modèle géométrique et le rendu de ce modèle. Le rendu englobe souvent l'extraction dynamique des textures pour le modèle en fonction du point de vue. L'extraction dynamique des textures implique qu'on ne connaisse pas les textures a priori et que l'extraction de ces dernières se fait à mesure que le point de vue dans lequel le modèle est rendu est connu. L'objet de ce travail concerne la deuxième étape et bien que le présent document est lui-même divisé en deux parties traitant de la reconstruction géométrique (chapitre 3) et de la reconstruction des textures (chapitre 4), il reste que moins d'importance sera accordée à la reconstruction géométrique qui est considérée a priori dans notre technique.

Comme indiqué plus haut, ce travail s'intéresse à l'extraction et l'unification des textures provenant d'images et plus particulièrement de photographies. Dans un premier temps, l'extraction des textures se fait à l'aide d'un modèle géométrique et des matrices de projection relatives à chaque image. Les techniques de reconstruction des modèles et d'obtention des matrices de projection des images sont brièvement exposées au chapitre 3. Notre technique de reconstruction est traitée en détails dans un docu-

ment connexe [Fra98]. L'échantillonnage des données d'une texture dans une image se fait en transformant l'échantillon tridimensionnel correspondant à l'élément de texture sur la facette par la matrice de projection de l'image. Le point dans l'image ainsi obtenu nous donne la couleur de la facette au point d'échantillonnage. Une fois le modèle géométrique et les matrices de projection reconstruits à partir des images, la phase d'extraction des textures pour chaque facette du modèle est automatique et ne requiert qu'une légère intervention de l'utilisateur<sup>1</sup> qui pourrait aisément être automatisée. Notre technique d'extraction des textures est décrite en détails à la section 4.2.

D'autre part, le fait que nous soyons en possession de nombreuses images d'une même scène<sup>2</sup> nous oblige à considérer des points de vue différents d'une même facette d'un modèle, donc d'une même texture. Les couleurs des images, qu'elles soient synthétiques ou photographiques, sont influencées par plusieurs facteurs dont plusieurs sont liés directement à la position et la direction du point de vue. Ainsi, puisqu'on désire obtenir une texture unique pour chaque facette du modèle géométrique, il faut discriminer et joindre ensemble les diverses couleurs provenant de points de vue différents de façon à créer une texture unissant les informations les plus précises et les plus cohérentes des images d'où elles proviennent. Cette tâche d'unification et de traitement des textures se fait à l'aide de plusieurs données provenant à la fois du modèle et des images.

D'une part, les images comportent des transformations particulières dues à la correction des couleurs, la précision et la résolution ainsi que les déformations de lentilles des caméras dont il faut tenir compte si on veut reconstruire des modèles complets sans égard spécial quant à la provenance des images utilisées. Nous entendons par modèles complets, des modèles qui contiennent toutes les informations nécessaires pour en faire un rendu réaliste. Dans notre cas, cette complétude implique une définition géométrique ou polygonale (sommets) et une définition texturale (image de couleurs paramétrisées sur la surface). Des détails relatifs à la scène représentée dans les images tels les occlusions entre primitives, les normales aux surfaces et la planarité des facettes sont aussi à analyser afin d'évaluer de façon relativement précise les divers éléments

---

<sup>1</sup>La section 4.3.1 explique la tâche d'alignement des textures qui est sous-entendue ici.

<sup>2</sup>Tout comme la vue binoculaire chez l'humain, les techniques de reconstruction requièrent normalement au moins deux images d'une scène pour pouvoir déduire les informations tridimensionnelles de cette dernière.

de texture (*texels*) extraits des images. Plusieurs aspects de l'unification sont reliés à des problèmes très complexes en vision et en traitement d'images et les techniques que nous avons utilisées impliquent l'intervention de l'utilisateur qui semble être en mesure de pallier aux nombreuses difficultés et erreurs encourues par une méthode de vision standard entièrement automatique. Les méthodes et les problèmes reliés à l'unification des textures sont discutés en profondeur dans la section 4.3.

Finalement, nous traiterons des problèmes soulevés par notre recherche et des solutions seront suggérées. Plusieurs détails restent irrésolus au niveau de l'unification des textures. Certaines techniques produisent de bons résultats mais il est souvent difficile de les corroborer à l'aide d'arguments entièrement objectifs. Évaluer la beauté d'une texture sera toujours, et ce fort heureusement, de l'ordre du subjectif et il nous est facile d'affirmer qu'à ce niveau, l'intervention humaine ne pourra jamais être remplacée.

## Chapitre 2

# Modélisation à partir d'images

Ce chapitre fait le survol des nombreuses techniques utilisant les images pour faire de la modélisation. La modélisation est un concept très vaste et dans notre cas, nous resterons au niveau de la modélisation pour des fins infographiques, ce qui ne laisse pas moins une multitude de facettes de ce domaine à traiter.

Une image est un moyen extrêmement versatile pour représenter et stocker des informations. Elle est très intuitive car elle est l'analogie directe du produit de notre sens le plus noble et le plus utilisé : la vue. L'idée de modéliser des objets par des images de ces derniers n'est certainement pas nouvelle. Elle constitue la base didactique d'une majorité de nos connaissances et tire son origine du mécanisme de la vue qui produit notre perception des objets réels en unissant les informations de deux images, images provenant de nos deux yeux. Prise unitairement, l'image est certainement le moyen le plus intuitif et le plus efficace pour véhiculer de l'information, surtout si cette information représente des objets possiblement perceptibles par la vue. Prises en groupe, les images constituent des systèmes d'informations complexes pouvant recréer le mouvement (films) et plus tacitement la tridimensionnalité (vue binoculaire, stéréoscopie) des objets représentés. Toutes les techniques de modélisation à partir d'images ressemblent beaucoup à la perception visuelle humaine en ce sens qu'elles obtiennent leurs informations de sources similaires, en l'occurrence des images et plus spécialement des photographies. Elles visent aussi les mêmes fonctionnalités telles la reconnaissance de la géométrie et des nombreuses propriétés des objets constituant une description "visuelle" des choses. Bien entendu, certaines données requises par les techniques de rendu infographiques, telles que les normales aux surfaces ainsi que les coefficients diffus et

spéculaires, s'éloignent un peu des données issues de notre perception, mais souvent, elles comportent de multiples ressemblances (on n'aura qu'à penser aux couleurs des textures par exemple).

La section 2.1 traite des techniques de modélisation à base d'images qui, par conséquent, ne font pas de reconstruction ou qui n'utilisent pas de modèles géométriques pour représenter les objets. Cette section englobe les domaines des extrapolations de vues et du rendu par images (*Image-Based Rendering* ou *IBR*). La section 2.2 expose les techniques axées sur l'utilisation de modèles géométriques et l'extraction d'informations inhérentes aux modèles contenus dans les images. Nous verrons brièvement les techniques de reconstruction des textures et des propriétés de surface dans les images. Une partie de cette section sera consacrée aux méthodes de rendu impliquées par ces techniques.

## 2.1 Interpolation de vues et rendu à base d'images

Pour des applications à vocation de rendu, on peut se servir des images pour extrapoler d'autres points de vue sans reconstruire de modèles texturés. En mettant l'emphasis sur la reconstruction des textures dans ce travail, le rendu devient un aspect essentiel pour justifier la nécessité des textures en plus des modèles géométriques. Conséquemment, il est important de discuter des techniques qui utilisent les images directement pour le rendu en opposition à notre technique qui implique la reconstruction d'un modèle géométrique. Même si les techniques de rendu basées sur les modèles géométriques texturés sont très répandues et offrent une grande liberté au niveau de la spécification et de la manipulation des modèles, le rendu à base d'images présente aussi des avantages significatifs quant à la grande complexité des objets modélisés d'où découle une grande partie de son réalisme.

Deux divisions particulières se distinguent dans ce domaine de recherche en infographie. Bien qu'elles soient toujours plus ou moins fondées sur l'échantillonnage et l'interpolation des données provenant d'images, les techniques de rendu à base d'images se regroupent en les techniques d'interpolation d'images de points de vue rapprochés et les techniques d'interpolation de champs de lumière paramétrisés.

### 2.1.1 Techniques de rendu par interpolation d'images

Les techniques de rendu par interpolation d'images créent une nouvelle image d'un nouveau point de vue en faisant la jonction des informations contenues dans les images environnant ce nouveau point de vue. Les images utilisées représentent des points de vue relativement rapprochés et disposés sur une trajectoire connue de sorte que les informations sont faciles à corréler et à interpoler entre elles. Le procédé de rendu évoqué par ces techniques est indépendant de la complexité de la scène représentée et peut s'appliquer à toutes sortes d'objets qu'il est possible de modéliser ou plus spécialement de photographier. Dans ce dernier cas, nous verrons que la complexité des modèles reconstruits par notre système est très faible en comparaison de la complexité des modèles réels pouvant être capturés par un appareil photo.

Certaines de ces techniques [Che95] [SS97] utilisent des projections cylindriques ou sphériques de l'entourage des objets pour produire des environnements d'images panoramiques projetées dans lesquels on peut naviguer selon des trajectoires particulières déterminées par les méthodes de projection. Par exemple, dans un environnement cylindrique, on dispose la caméra au centre du cylindre nous permettant de tourner autour de l'axe central. Les navigations peuvent être diversifiées en disposant plusieurs environnements cylindriques adjacents et en permettant des sauts d'un environnement à l'autre. La figure 2.1 illustre deux types d'environnement de projection. Les images de droite dans cette figure sont tirées de la recherche de McMillan et Bishop [MB95] sur la modélisation d'environnements plénoptiques. Plus fondamentalement, Lippman [Lip80] avait proposé la génération d'une image de chaque point de vue principal dans une scène pour une exploration réaliste dans cette dernière, sans toutefois intégrer les environnements de reprojection plus ou moins complexes impliqués dans les techniques présentées plus haut.

D'autres techniques ne forment pas d'environnements globaux de projection et interpolent directement deux images rapprochées pour créer une transition uniforme entre les deux points de vue différents. Quelques techniques de mutation des images sont présentées, entre autres par Beier et Neely [BN92] et Seitz et Dyer [SD96]. Malheureusement, ces techniques exigent souvent une mise en correspondance manuelle des images de sorte qu'elles requièrent un effort au moins aussi considérable que celui impliqué dans la construction des environnements de projection des techniques précédentes. De plus,

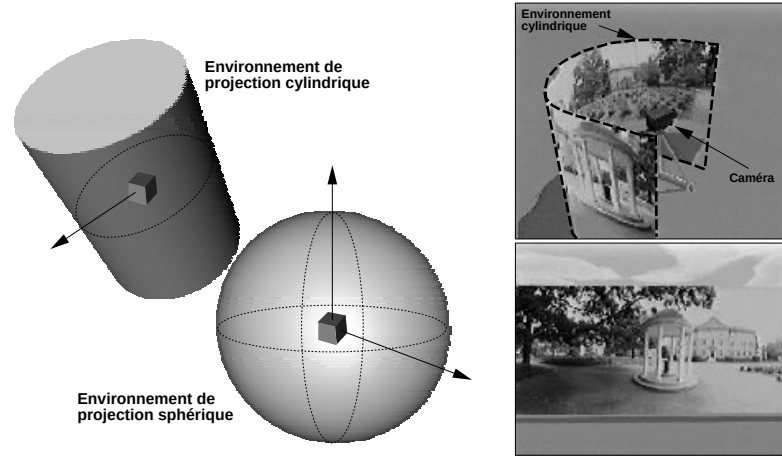


FIG. 2.1: Les environnements de projection cylindrique et sphérique.

de façon générale, ces mutations ne sont pas géométriquement correctes ce qui leur enlève une part de leur pertinence dans l'analyse des techniques de rendu réalistes.

### 2.1.2 Techniques de rendu par champs de lumière

D'autres techniques utilisent les informations des images pour construire des champs de lumière (radiances). Ces champs sont paramétrisés de diverses façons et encodent la lumière de façon directionnelle de sorte qu'on échantillonne le champ selon un certain point de vue pour obtenir une nouvelle image de l'objet illuminé dans ce point de vue. Le champ de lumière construit par ces techniques est caractérisé par la fonction plénoptique simplifiée à quatre paramètres. En fait, la fonction plénoptique exprime la radiance en un point de l'espace pour une direction donnée et comporte normalement cinq paramètres. Puisque pour une direction donnée, la radiance ne change pas le long de cette dernière, on peut réduire la fonction plénoptique à quatre dimensions. Le champ de lumière discret d'un objet doit encoder un bon nombre de radiances de sorte qu'on puisse échantillonner le champ avec une bonne résolution. Un des désavantages majeurs des techniques qui utilisent les champs de lumière réside dans l'occupation mémoire du grand nombre de données. On doit effectivement stocker une énorme quantité d'informations (des centaines de mega-octets pour un champ simple) pour pouvoir représenter les objets sous plusieurs points de vue différents. Tout dépendant de la paramétrisation et de la méthode de stockage des données, le champ de lumière consiste en quelque sorte à l'ensemble des traces ou images de radiances d'un objet provenant d'un grand

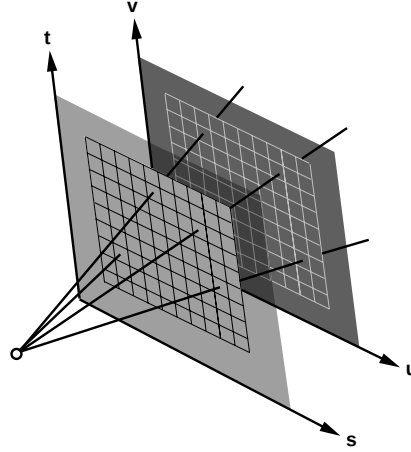


FIG. 2.2: Paramétrisation des rayons d'échantillonnage d'un champ de lumière par deux plans.

nombre de points de vue de ce dernier.

Le stockage des données du champ dépend directement de la paramétrisation choisie qui détermine la façon dont on échantillonne la radiance de l'objet. Levoy et Hanrahan [LH96] et Gortler et al. [GGSC96] proposent une paramétrisation à deux plans pour l'échantillonnage des rayons autour de l'objet. La figure 2.2 nous montre comment l'échantillonnage des radiances est effectué dans une paramétrisation à deux plans. Comme on peut le constater, chacun des éléments du plan  $st$  aura la résolution du plan  $uv$ , c'est-à-dire  $u \times v$  données. C'est l'équivalent d'échantillonner  $s \times t$  images de résolution  $u \times v$  nous donnant en fait  $s \times t \times u \times v$  radiances à stocker. Tout récemment, Camahort et al. [CLF98] ont présenté une paramétrisation à deux sphères pour l'échantillonnage des champs de lumière.

Les techniques de rendu à base d'images sont fondamentalement dédiées à la production d'images réalistes sans la nécessité de posséder un modèle géométrique complexe. Cette particularité du domaine lui confère une importance particulière car il englobe à la fois les aspects de modélisation et de rendu. En fait, nous ne sommes plus aux prises avec un procédé infographique unique, scindé en la modélisation d'une scène et le rendu de cette dernière. La modélisation de la scène est véritablement devenue le rendu de celle-ci et vice-versa. Entre autres, dans la majorité des cas en infographie, il a toujours été naturel de considérer les modèles synthétiques par leurs représentations imagées, leur rendu infographique. Bien entendu, certaines manipulations et trans-

formations géométriques des modèles sont freinées par l'inexistence d'une abstraction géométrique, mais en considérant le but tacite de l'infographie, qui est de produire des images synthétiques réalistes, les techniques de rendu à base d'images proposent des solutions directes et efficaces à une multitude de problèmes infographiques s'adressant directement au réalisme.

## 2.2 Projection de textures

Un des avantages majeurs de l'utilisation d'images et particulièrement de photographies pour le rendu est relié au réalisme de ces dernières en fonction de l'efficacité des techniques. Les techniques de rendu standards en temps réel telles qu'utilisées dans les jeux et dans les systèmes d'animation requièrent tout particulièrement l'aide des textures pour le réalisme. Les techniques efficaces de rendu tirent souvent profit de méthodes d'affichage tel le *z-buffer* et d'illumination tel le *Gouraud shading* qui sont supportées par des modules matériels. Les accélérations matérielles sont aussi très importantes au niveau du traitement (transformations, appositions et pré-filtrage) des textures. Les photographies constituent certainement des représentations extrêmement réalistes d'un modèle et l'apposition des textures extraites de ces photographies sur ce dernier peut donner, dans une bonne majorité des cas, des rendus impressionnants et ce de façon très efficace sans avoir à utiliser les techniques de rendu comme le lancer de rayons et l'illumination globale par radiosité.

Dans cette section, nous considérons l'existence du modèle géométrique construit a priori et ayant plus ou moins d'influence sur les données à extraire pour le rendu de ce dernier. Le chapitre 3 concerne plus particulièrement la reconstruction géométrique à partir d'images.

Ceci nous amène à parler des techniques de rendu avec textures dépendantes du point de vue. Tout comme les techniques discutées à la section 2.1, un rendu dépendant d'un point de vue compose les images sources et interpole les données de façon à produire l'aspect le plus réaliste possible d'une scène en fonction du point de vue donné. La différence est qu'on interpole les textures qui sont apposées sur les modèles au lieu de générer une image complète sans structure géométrique sous-jacente comme dans les techniques de rendu à base d'images vues précédemment.

Par ce travail, nous visons l'obtention d'un modèle complet afin d'en faire le rendu

a posteriori. Ceci influence beaucoup les différentes méthodes utilisées pour reconstruire la géométrie et surtout les textures. Les modèles complets que nous obtenons peuvent servir à une foule d'applications dont le rendu. Comme nous allons le voir dans la conclusion de ce travail (voir le chapitre 5), la généralité de nos modèles impose des exigences sévères sur la qualité des textures, exigences que les techniques qui seront présentées dans cette section n'ont pas car elles sont principalement dédiées à faire un rendu "spécialisé" basé directement sur les images sources.

Certains systèmes de reconstruction présentés au chapitre 3 font le rendu de leur scène reconstruite à partir d'images à l'aide d'une technique de projection des images sur le modèle ou d'échantillonnage dynamique des textures. Parfois, ils composent directement les textures sur le modèle selon le point de vue donné. Cette façon de faire ressemble un peu aux techniques de rendu à base d'images présentées à la section 2.1 avec cette particularité que le résultat du rendu nous donne une texture qui est apposée sur le modèle au lieu d'une image complète de la scène projetée. Segal et al. [SKv<sup>+</sup>92] ont présenté une technique efficace d'illumination et d'ombrage de scènes par projection de textures. Cette technique de rendu ouvrait la porte à plusieurs autres techniques utilisant la projection d'images ou de textures sur les modèles pour simuler des phénomènes complexes d'illumination de façon efficace et réaliste.

Le système *Façade* [DTM96] produit des modèles géométriques de façon interactive en laissant le soin à l'utilisateur de modéliser la géométrie tridimensionnelle et de l'associer avec les images de cette dernière (voir la section 3.3). Une fois la scène géométrique reconstruite, le rendu de cette dernière est effectué directement en projetant et en interpolant les images sur le modèle selon un point de vue particulier. Bien qu'il soit avantageux pour la reconstruction géométrique impliquée par cette technique que les images de la scène soient relativement distancées et en petit nombre, pour le rendu par contre, ce fait comporte quelques désavantages. D'une part, la parcimonie des points de vue ayant servis pour construire les modèles causera des "trous" dans les textures pour certains points de vue arbitraires, ce qui implique une extrapolation des informations pour combler ces "trous". D'autre part, la variété des points de vue pouvant être utilisés pour la reconstruction géométrique peut provoquer des chevauchements entre les images soumises aux différents phénomènes exposés à la section 4.3.2. Ces chevauchements de données fournissent plusieurs valeurs pour un élément de texture donné.

Conséquemment, ces valeurs devront être interpolées sur le modèle ; on les joint à cet effet en fonction de la distance et de la direction de vue (voir la figure 4.16 de la section 4.3.2). Plusieurs améliorations, plus spécialement au niveau de la génération d'un environnement de projection des textures et d'unification des chevauchements dans ces dernières ainsi qu'au niveau du remplissage des “trous”, ont été apportées à cette technique de rendu par projection de textures [DBY98] qui ne demande, en somme, aucun calcul d'illumination coûteux puisqu'on se sert directement des couleurs des images qui sont habituellement des photographies ou des images de rendu.

Havaldar et al. [HLM96] calculent une transformation projective des triangles de la scène reconstruite dans les images pour les échantillonner. La “meilleure” image à échantillonner pour un triangle donné est déterminée en fonction de l'aire de projection du triangle dans l'image. Contrairement à la technique de Debevec et al. [DTM96], on projette les primitives dans les images avec une transformation projective d'image à image et non de monde à image. Ceci provoque quelques problèmes quant à la correction des différentes perspectives des images se retrouvant dans les textures sans compter les calculs de visibilité et d'occlusion. Niem et Broszio [NB95] utilisent une technique similaire à celle du système *Façade* (angle et distance) pour déterminer la “meilleure” image pour effectuer l'échantillonnage des textures pour une primitive donnée. Puisque deux primitives adjacentes peuvent supporter des textures provenant d'images différentes, une technique de mélange des texels frontaliers est utilisée pour diminuer les contrastes dus aux différences dans les images d'où proviennent les textures. Les phénomènes pouvant causer les différences dans les images de différents points de vue sont expliqués à la section 4.3.2.

Globalement, dans les techniques de rendu à base d'images avec modèles géométriques, plus les textures ou les images sont complexes moins les modèles géométriques sous-jacents requièrent une grande précision et une grande complexité. Si on les utilise comme supports à un procédé d'apposition de textures et de rendu dynamique basé sur les textures, les modèles géométriques réalistes ne sont plus indispensables pour des fins infographiques. Par contre, comme nous allons le voir dans le chapitre 3, le caractère tangible des modèles géométriques ainsi que leur description simple et versatile font qu'ils occuperont encore longtemps une place de choix dans la représentation des scènes synthétiques en infographie. L'avènement de nou-

velles techniques de reconstruction efficaces et complètes comme la nôtre contribuera certainement à étendre les applications des modèles géométriques dans un domaine où la progression et l'expansion effacent les frontières et contribuent à la création de techniques infographiques de plus en plus hybrides formant des systèmes de rendu à grande envergure.

## Chapitre 3

# Reconstruction géométrique

Dans ce chapitre, nous décrivons brièvement le vaste domaine de la reconstruction géométrique à partir d'images et plus particulièrement à partir de photographies. Pour considérer la pertinence de l'utilisation des images dans la modélisation géométrique nous croyons utile de voir ces dernières comme des plans, des guides sur lesquels on peut baser la construction des modèles. Avant l'utilisation directe des images dans les systèmes de reconstruction, on pouvait dire que la tâche de modélisation consistait en l'agencement de blocs de construction dans un référentiel plus ou moins bien défini. D'une part, la plupart des systèmes de modélisation sont très libéraux quant à la spécification des dimensions et des positions des primitives dans la scène. De plus, l'utilisateur de tels systèmes dispose souvent les éléments d'une scène plus ou moins précisément dans un espace tridimensionnel souvent sans dimension finie ou limite apparente. Lorsqu'on avait une idée, quelques dessins ou des photographies d'une scène ou d'un objet, il incombait, et il incombe toujours, à l'utilisateur de tels systèmes d'essayer de traduire le tout dans un système de modélisation et ce, de façon plus ou moins précise et efficace. À moins qu'il ne possède un extraordinaire sens des distances et des proportions, l'utilisateur est contraint d'établir des points de repers imprécis ou d'avoir recours aux mesures prises sur le terrain, ces dernières étant souvent inexistantes.

Comme nous allons le voir, l'utilisation des photographies évite à l'utilisateur beaucoup de travail relié à l'établissement des proportions d'un modèle et à sa situation spatiale dans une scène. De plus, elle permet une reconstruction avec une grande précision, précision s'approchant nettement de celle pouvant être atteinte par les utilisateurs les plus doués des systèmes de modélisation standards.

Dans les sections suivantes, nous présenterons les techniques courantes de reconstruction à partir d'images. Nous avons distribué ces techniques en deux grandes catégories basées sur l'interactivité, soient les techniques de reconstruction dites *automatiques* (section 3.2) et celles dites *interactives* (section 3.3). La section suivante (section 3.1) donne une brève définition de la reconstruction géométrique à partir d'images et discute des problèmes communs aux différentes approches de reconstruction. Subséquemment, la section 3.4 expose en détails la technique de reconstruction géométrique que nous avons utilisée dans ce travail.

### 3.1 Mise en situation

La reconstruction géométrique à partir d'images consiste en l'établissement de corrélations entre des images d'une scène et la scène elle-même. Ces corrélations se retrouvent habituellement sous forme matricielle et l'utilisation de la géométrie projective est très bien adaptée au problème de reconstruction. L'énoncé du problème est simple en fait : étant donné un ensemble d'images d'une scène de différents points de vue, on veut reconstruire la scène de façon à obtenir une représentation surfacique utile pour la manipulation et le rendu de cette dernière. Si les points de vue des images diffèrent peu, il est facile d'établir automatiquement les correspondances entre les images adjacentes mais il est plus difficile d'inférer quelconques informations géométriques (profondeurs) à propos de la scène concernée. L'inverse est aussi vrai. La possession d'images éloignées et très différentes nous permet de déduire plus facilement et plus précisément l'information  $3D$  mais la mise en correspondance automatique des images est toutefois beaucoup plus difficile dans ce cas que pour les images plus rapprochées et similaires. De plus, il est important que les points de vue soient variés au niveau directionnel et pas seulement au niveau de la position. Des images représentant une scène à différentes distances mais toujours dans la même direction n'offrent pas beaucoup d'informations pertinentes sur la scène tridimensionnelle. Comme énoncé plus haut, un avantage marqué de l'utilisation des images en modélisation c'est qu'elles offrent une base, des limites et des contraintes de construction qui sont souvent d'un grand secours pour établir efficacement la position, la taille et la forme des objets dans une scène. Toutes les techniques de reconstruction à partir d'images, qu'elles soient automatiques ou non, s'établissent généralement sur le principe de correspondances entre les images. L'information tridi-

mensionnelle manquante dans les images est déduite de ces correspondances et l'aspect automatique ou manuel des systèmes de reconstruction à partir d'images découle habituellement du fait de l'établissement automatique ou manuel de ces correspondances.

La modélisation standard sans image comporte de nombreux problèmes telle la difficulté pour l'utilisateur de se situer et de manipuler des objets dans un espace virtuel plus ou moins intuitif. La métaphore entre le réel et le système de modélisation peut parfois être ambiguë, mais on manipule directement des objets tridimensionnels complets dans de tels systèmes contrairement à l'effort de déduction requis dans les systèmes de reconstruction à partir d'images. Certes, les images offrent un moyen direct d'établir la géométrie des objets dans une scène mais nous croyons toutefois que les projections d'une scène et les mises en correspondance entre ces projections ne constituent pas nécessairement une tâche plus facile que de manipuler directement un objet tridimensionnel dans un environnement plus ou moins rapproché de la réalité.

La majorité des systèmes de reconstruction à partir d'images exige la calibration des images, qui implique le calcul des diverses informations relatives aux caméras qui se fait prioritairement à la reconstruction en tant que telle. Dans la majorité des cas, la position et la direction de vue pour chaque image sont des informations suffisantes pour la définition d'une caméra. Cette étape initiale est souvent caractérisée par l'entrée de points dont on connaît la position tridimensionnelle exacte et qui donneront les premières correspondances "stables" du système. Évidemment, plus le nombre de points entrés (manuellement) est grand et leurs coordonnées précises, plus la reconstruction des autres points de la scène sera précise. Certains systèmes demandent un grand nombre de points de départ. L'initialisation de la plupart des systèmes de reconstruction à partir d'images demande de six à huit points pour donner de bons résultats. D'autres systèmes exigent régulièrement l'entrée de beaucoup plus de points. De plus, ces points d'initialisation doivent satisfaire certaines exigences. Ils doivent fournir des informations sur les trois dimensions à la fois dans l'espace (i.e. ils ne sont pas coplanaires) et une fois projetés dans les images. Les points tridimensionnels qui projettent en un seul point dans une image aident peu à retrouver cette information. Trouver les bons points qui projettent correctement dans chaque image et qui donnent des résultats précis est une tâche impliquant souvent des mesures sur la scène réelle ou l'utilisation d'objets simples et faciles à calibrer.

Pour les reconstructions effectuées avec notre système pour ce travail, nous avons souvent utilisé un cube dans les images pour faire l'initialisation (voir la section 3.3). Les sommets du cube sont faciles à identifier et aucune mesure n'était nécessaire puisque nous n'avions qu'à définir un seul sommet pour l'origine et à établir une longueur d'arête pour dériver instantanément les autres sommets. Si le cube n'est pas mesuré directement sur la scène, ce qui est habituellement le cas, la reconstruction s'effectue à un facteur d'échelle près pour chaque dimension. Comme expliqué par Frasson [Fra98], il est aussi préférable d'avoir une distribution uniforme des points de calibration sur l'image et dans l'espace pour obtenir de meilleurs résultats. De plus, il est difficile de borner les erreurs et d'identifier leurs causes véritables quand elles proviennent de l'initialisation. L'utilisateur est laissé avec certaines spéculations quant à la validité des points entrés, qui influence directement la précision de la reconstruction. On constate finalement que cette tâche d'initialisation du système comporte tous les inconvénients propres aux techniques de modélisation standards que nous voulons idéalement éviter par la modélisation à partir d'images.

Tout ceci soulève un grand point que l'analyse des techniques de reconstruction pourra peut-être éclaircir. L'établissement de correspondances entre les images constitue une tâche beaucoup plus servile et aveugle que celle de manipuler des objets directement dans la scène. Elle comporte, comme nous allons le voir, des problèmes différents mais tout aussi importants que ceux impliqués dans la modélisation standard. Il est difficile de comparer les techniques de reconstruction à partir d'images à celle de modélisation sur les mêmes bases d'argumentation. La reconstruction est souvent plus simple, plus précise et plus efficace avec des images que sans, même si on néglige le temps impliqué dans l'initialisation du système. Dans la plupart des activités de reconstruction et ce, dans la majorité des systèmes, cette étape d'initialisation représente une infime partie de l'effort en comparaison de celle occupée par modélisation de primitives  $2D$  et par la mise en correspondance de ces primitives entre les images. La modélisation d'objets complexes et de scènes complètement imaginaires est une tâche plus facile dans un système de modélisation standard que dans un système de reconstruction avec images. Dans ce dernier cas, nous sommes limités par la complexité et la qualité des informations contenues dans les projections (images) des objets, et aussi par la rareté et souvent l'inexistence, si on considère des photographies, d'images

d'objets irréels ou imaginaires. Certaines techniques [FR92][DTM96] utilisent des simulations de points de repères pour identifier les ressemblances entre certains objets réels et imaginaires mais les comportements géométriques de ces derniers doivent cependant rester très simples (une quadrique naturelle par exemple). Ces techniques sont sujettes à une foule de problèmes reliés à la topologie des modèles tels les problèmes d'auto-pénétrations de la surface de ces derniers. Dans tous les cas, ces techniques constituent de nombreuses limites au vaste étendu imaginaire pouvant possiblement se traduire dans les systèmes de modélisation standards.

## 3.2 Méthodes épipolaires

L'utilisation d'images dans la modélisation de scènes géométriques évoque plusieurs automatismes qui simplifient grandement la tâche de reconstruction. Plusieurs de ces automatismes proviennent du domaine de la vision informatisée. En fait, un des buts fondamentaux de la vision informatisée consiste en cette démarche inverse à celle de l'infographie qui veut qu'on obtienne des informations inhérentes à la scène (virtuelle ou réelle) à partir d'images de cette dernière. La démarche infographique, qui se concentre surtout sur la synthèse d'images de modèles virtuels, s'éloigne un peu des principes sous-jacents à ceux impliqués dans la modélisation ou la reconstruction géométrique à partir d'images. Étant donnée cette proximité conceptuelle avec la vision informatisée, la reconstruction à partir d'images trouve de nombreux champs d'intérêts et de recherche dans ce domaine. Cette section constitue une revue des méthodes de reconstruction à partir d'images, qui sont basées sur des techniques directement reliées à la vision informatisée.

La vision informatisée évoque habituellement un certain caractère automatique. Que ce soit dans des sphères de recherche particulières comme la détection de contours ou la reconnaissance de formes dans les images, l'intervention de l'utilisateur dans les systèmes de vision est habituellement négligée au profit d'heuristiques et d'algorithmes autonomes.

### 3.2.1 Recherche sur les lignes épipolaires

Les techniques de vision automatiques utilisent généralement la géométrie épipolaire pour effectuer la recherche des correspondances entre les images. La figure 3.1 illustre l'approche de recherche sur une ligne épipolaire.

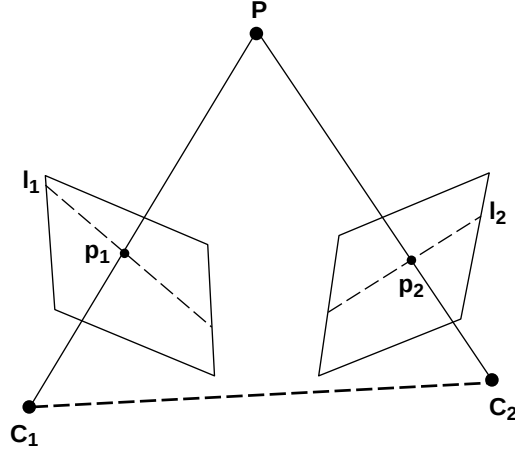


FIG. 3.1: La recherche pour la correspondance  $p_1 p_2$  étant donné  $p_1$  se fait sur la ligne épipolaire  $l_2$ .

Étant données deux images dont les points de vue ou caméras sont  $C_1$  et  $C_2$  et un point tridimensionnel  $P$  dont on veut associer la projection dans les deux images, on trouve le plan  $C_1 C_2 P$ . Ce plan épipolaire intersecte les images en deux lignes épipolaires,  $l_1$  et  $l_2$ . Nous constatons que la projection  $p_1$  et  $p_2$  de  $P$  dans chaque image se trouve sur les lignes épipolaires respectives. Ainsi, si  $p_1$  est donné alors  $p_2$  peut être recherché automatiquement sur la ligne épipolaire  $l_2$ . Le problème de recherche bidimensionnelle de couleurs dans les images se réduit donc à un problème de recherche unidimensionnelle de couleurs sur les lignes épipolaires.

La géométrie épipolaire nous permet de trouver plus efficacement les correspondances entre les images pour ainsi calculer la position tridimensionnelle des points composant la scène à reconstruire. La théorie englobant le calcul des points tridimensionnels à partir des correspondances obtenues entre les images est expliquée à la section 3.4.

La détermination automatique des correspondances entre les images à l'aide de la technique de géométrie épipolaire implique la sélection de points précis dans les images<sup>1</sup> qu'il faut chercher sur les lignes épipolaires correspondantes dans les autres images. Le sujet de cette section se concentrant sur les techniques de vision automatiques pour la reconstruction, il importe donc d'exposer les techniques de vision permettant la sélection automatique des points dans les images sur lesquelles il faut effectuer la re-

<sup>1</sup>Le point  $p_1$  dans la figure 3.1 a nécessairement dû être fourni d'une quelconque façon pour effectuer la recherche.

cherche. Nous discuterons donc des techniques de détection de contours dans les images comme étape initiale à la recherche épipolaire décrite plus haut.

Plusieurs méthodes en vision nous permettent d'identifier les contours des objets dans les images. Nous élargissons ici la définition du concept de détection de contours dans les images. Nous entendons par détection de contours non seulement l'identification des lignes ou segments de lignes dans une image, mais aussi la détection des points formant les coins des objets et la détection des formes générales représentées dans les images. Gonzales et Woods [GW92] présentent plusieurs techniques standards en vision pour la détection de diverses informations pertinentes dans les images (contours, coins, etc.). Pour simplifier l'algorithme de recherche épipolaire on voudra bien entendu effectuer la détection sur les éléments frontaliers des objets tels les coins de ces derniers. La représentation des scènes privilégiée dans les systèmes de reconstruction étant la représentation frontalière, il est en effet plus redondant de faire une recherche épipolaire pour tous les points d'une image alors que la description des modèles utilisée nécessite seulement les sommets des objets. Cependant, il pourrait être utile, dans une représentation plus générale, d'effectuer la détection sur les textures des facettes dont certains sommets sont cachés ou moins précis que quelques détails visibles sur sa surface. Il faut tenir compte toutefois que les points contenus sur une surface sans texture contrastée sont habituellement beaucoup moins discernables que ses sommets dans les images. L'établissement de correspondances à partir des textures devient cependant indispensable si on est aux prises avec des surfaces courbes dont la silhouette change d'une image à l'autre.

Le système *AIDA* [WG96] est un système de reconstruction entièrement automatique qui utilise une banque de contraintes pour optimiser l'algorithme de reconnaissance de formes dans les images. Comme nous allons le voir dans la section suivante, les algorithmes de vision automatisés composent souvent leurs actions en s'aidant d'hypothèses sur les images et sur la nature de la scène qui y est projetée. Bien qu'il ne soit pas directement fondé sur la mise en correspondance automatique par recherche épipolaire, le système *AIDA* est sujet aux mêmes hypothèses et plusieurs erreurs peuvent être introduites lors de la sélection automatique des contraintes à appliquer sur les modèles. La section 3.3 discute en détails des techniques de reconstruction basées sur les contraintes qui requièrent l'intervention d'un usager.

### 3.2.2 Problèmes de la technique épipolaire

Bien que la technique de recherche sur une ligne épipolaire offre plusieurs avantages intéressants, surtout au niveau de la simplicité et de l'efficacité, elle recèle de nombreuses difficultés qui mettent en péril son utilisation pertinente dans le système impliqué dans ce travail. Plusieurs phénomènes complexes interviennent dans l'apparence des images et les objets apparaissant dans ces dernières peuvent changer radicalement d'aspect d'un point de vue à l'autre. La recherche sur une ligne épipolaire suppose qu'il existe un rapprochement plus ou moins direct entre deux pixels de deux images et est donc très sensible à la ressemblance des deux images concernées.

Une occlusion dans une image garantira que la recherche échouera ou donnera une correspondance erronée à défaut de trouver la couleur attendue qui est cachée. Le pixel recherché comportant une réflexion spéculaire sera possiblement de couleur différente que la couleur recherchée et causera les mêmes ennuis. Sans l'aide de l'utilisateur, il est extrêmement difficile, voire même impossible de déterminer si le point recherché se trouve bel et bien dans une image. Si les images sont des photographies, elles seront sujettes à la correction de couleur des appareils photo, au bruit, à la discrétisation et au mélange des couleurs dans les pixels, aux effets de focalisation, etc., tous des phénomènes très complexes à simuler et qui causent des différences souvent notables entre les couleurs des images et qui feront échouer la recherche. Entre autres, les contours des polygones impliqués dans l'extraction étant infiniment fins, il y a forcément un mélange entre les couleurs de l'arrière-plan et les couleurs du polygone dans un pixel donné. Il est facile de remarquer qu'il est excessivement rare d'être en présence d'aucun mélange de couleurs sur les bordures d'un polygone. C'est pourtant ces mêmes contours que les algorithmes de vision automatisés détectent le plus souvent.

Finalement, la configuration spatiale des images influence l'algorithme non seulement parce qu'elle provoque des différences de couleurs dans les images (réflexions spéculaires) mais aussi parce que certaines configurations spatiales des images peuvent ne pas comporter de plan épipolaire pour un certain point. Par exemple, les points  $P$ ,  $C_1$  et  $C_2$  peuvent être colinéaires dans la figure 3.1. De plus, cette technique suppose une calibration précise des images pour bien identifier les points focaux.

Pour ce qui est des algorithmes de détection de contours en particulier, l'étude des nombreux articles dans ce domaine et notre expérience dans l'implantation de cer-

tains outils de détection nous ont montré la spécialisation marquée des algorithmes à certains types d'images très précis. La plupart des techniques sont très sensibles à une paramétrisation complexe donnée par l'utilisateur et cette dernière doit être modifiée fréquemment selon les différentes images sur lesquelles elles agissent. Ce qui ressort de notre analyse de ces techniques est qu'elles fonctionnent bien mais requièrent une assistance humaine non négligeable. Elles nécessitent l'aide d'un utilisateur pour l'identification des résultats erronés causés par le contraste précis d'une ombre sur une surface par exemple; ce contraste qui engendre la détection de la frontière de l'ombre qui ne constitue pas une partie de la géométrie à reconstruire. L'utilisateur doit aussi discriminer les résultats inutiles tels les produits de la détection de contours sur une surface dont la géométrie pourrait être remplacée par une texture et que cette dernière comporte des motifs bien définis (des briques par exemple). Enfin, c'est à l'utilisateur de faire les diverses considérations quant à l'applicabilité des techniques pour un certain type d'images ou pour la paramétrisation continue de ces dernières pour qu'elles produisent de bons résultats pour une image donnée. Ce dernier point fait entrer en jeu des connaissances mathématiques subtiles sur la technique que la majorité des utilisateurs n'ont pas. Les techniques de vision ne fonctionnant pas bien en général pour tous les types d'images indifféremment, elles trahissent un précepte important de la philosophie de notre système de reconstruction qui veut utiliser toute l'information pertinente des images peu importe d'où elles proviennent et comment elles ont été obtenues.

En observant ces procédés de reconstruction utilisant des techniques automatiques, il est facile de constater l'importance considérable encourue par les erreurs, surtout au niveau géométrique. Il ne faut pas oublier que le but premier de la reconstruction géométrique telle que nous la voulons dans ce projet est d'obtenir des modèles relativement complexes et précis pour en faire un rendu réaliste. Plusieurs autres applications telle la détection de collisions par exemple requiert aussi la reconstruction de modèles précis. La notion de précision visuelle peut cependant s'éloigner de la notion de précision "robotique".

Comme nous avons pu le remarquer, les nombreuses erreurs se glissant dans les techniques automatiques peuvent compromettre sérieusement la cohésion et la précision des modèles reconstruits. Les modèles géométriques obtenus par reconstruction à partir d'images font face aux mêmes exigences quant à la robustesse et la précision que

les modèles obtenus par modélisation manuelle standard. La robustesse d'un modèle concerne des aspects géométriques importants pour plusieurs applications infographiques. Entre autres, la planarité, l'adjacence des faces (on veut éviter les fentes entre les faces) et l'ordonnancement des sommets d'un polygone ou tout facteur produisant des aberrations dans le modèle et mettant en péril le bon fonctionnement des applications infographiques l'utilisant, influencent négativement la robustesse d'un modèle. Ces exigences sont difficilement comblées par les techniques de vision automatiques qui intègrent mal l'intervention correctionnelle de l'utilisateur dans le processus de reconstruction. Dans la majorité des cas, l'ordinateur produit un nombre considérable d'erreurs et l'inaptitude de l'utilisateur à traiter et à corriger un tel nombre d'erreurs rend les techniques automatiques de reconstruction plus ou moins utiles pour la création de modèles à vocation infographique. Le nombre d'erreurs considérable et la subtilité de certaines de ces dernières font qu'elles sont difficiles à être toutes détectées nous donnant peu d'espoir d'obtenir des modèles très robustes dans le sens infographique du terme<sup>2</sup>

### 3.3 Méthodes par contraintes

Comme nous l'avons vu dans la section 3.2, bien que plusieurs techniques de vision sont bien adaptées aux problèmes de reconstruction à partir d'images, ces dernières pourraient souvent profiter de l'assistance de l'utilisateur pour corriger les nombreuses erreurs causées par une multitude de facteurs reliés, entre autres, à la qualité et la variabilité des images utilisées. Dans la plupart des cas les techniques automatiques de vision ne permettent pas l'intégration simple de l'utilisateur dans le protocole solutionnaire et souvent, les tâches de correction des erreurs a posteriori par l'utilisateur sont souvent complexes et inefficaces. Cette section traite des techniques de reconstruction adoptant des approches plus interactives qui sacrifient une grande partie de l'autonomie des techniques de vision pour donner un contrôle plus versatile à l'utilisateur. Notre système de reconstruction est basé sur ce principe d'interactivité plus polyvalent. En plus de déterminer efficacement les correspondances pertinentes à la reconstruction dans les

---

<sup>2</sup>Le manque de robustesse peut être provoqué par le dédoublement de sommets qui devraient être les mêmes causant ainsi une disjonction des faces d'un modèle par exemple. Mentionnons également les interpénétrations de la surface d'un modèle et la planarité des faces polygonales qui sont aussi des facteurs ayant une grande influence sur la robustesse des modèles.

images, l'utilisateur peut définir des contraintes géométriques sur le modèle qui permettent une convergence plus rapide des objets de la scène vers un état stable et précis.

Depuis quelques années déjà, des systèmes de reconstruction tels que *Façade* [DTM96], *PhotoModeler* [Pho] et *REALISE* [LdlFK<sup>+</sup>96], permettent de créer des scènes à partir d'images de façon interactive en intégrant avantageusement les interventions de l'utilisateur pour spécifier les correspondances et les contraintes.

Dans le système *Façade*, l'utilisateur construit un modèle tridimensionnel avec différentes primitives paramétrisées (boîtes rectangulaires, surfaces de révolution, etc.) de façon traditionnelle. Il établit par la suite les correspondances et les contraintes entre les primitives et les images. La description des modèles paramétrisés ou blocs constitue un encodage hiérarchique très efficace des contraintes. Les modèles sont spécifiés en terme de lignes ou de segments dans les images et les associations entre les blocs 3D et les images sont établies sur ces segments. Le système de contraintes est solutionné pour minimiser une fonction objective entre les modèles tridimensionnels et leur projection dans les images de façon non-linéaire. Dans ce système, les images sont utilisées pour corriger les modèles tridimensionnels construits par l'utilisateur de sorte qu'on pourrait l'évoquer comme un système de modélisation assistée par des images.

L'utilisateur du système *REALISE* spécifie quelques correspondances initiales entre les images pour ensuite laisser le système établir automatiquement les autres correspondances. Les interactions de l'utilisateur ne suffisent pas à soustraire ce système des problèmes soulevés à la section 3.2 à propos des méthodes de vision automatisées. Bien que les interventions correctionnelles de l'utilisateur sont suffisamment bien intégrées dans le processus pour représenter une forme d'assistance directe du système automatique, la reconstruction avec ce système est moins efficace qu'avec les autres si ce n'est que pour le temps dédié à la recherche épipolaire et aux corrections qu'elle engendre. Les caméras des images et les divers calculs de reconstruction sont solutionnés par une méthode non-linéaire.

*PhotoModeler* [Pho] est un système de reconstruction commercial fondé sur un mécanisme de reconstruction géométrique similaire à notre système. L'utilisateur entre des points dans les images et identifie les correspondances entre ces dernières. Le système de correspondances est résolu par une méthode non-linéaire. Les modèles simples reconstruits sont triangularisés et l'extraction des textures se fait directement dans les

images sans étape de correction et d'unification des informations.

### 3.4 Matrices de projection

Le système de reconstruction géométrique concerné par notre travail comporte certaines particularités qui le distinguent relativement des autres systèmes de reconstruction. Notre système est fondé sur le calcul d'une matrice de projection pour chaque image et la reconstruction des modèles  $3D$ , toutes basées sur la résolution d'un système d'équations linéaires. D'une part la matrice de projection pour chaque image ne constitue pas une matrice de projection standard et on ne peut pas nécessairement en déduire les paramètres exacts de transformations d'une caméra normale (translation, rotation, changement d'échelle, etc.). L'utilisation de cette matrice, des éléments de géométrie projective qui la concerne ainsi que des contraintes sur le modèle est répandue à travers tout le processus de reconstruction.

Cette section présente en détails les diverses étapes de reconstruction géométrique de notre système en mettant l'accent sur l'importance de la matrice de projection obtenue pour chaque image. Comme nous allons le voir au chapitre 4, ces matrices servent non seulement à la reconstruction géométrique des modèles, mais sont indispensables à l'extraction et à l'unification des textures. Les divers détails concernant l'utilisation des matrices de projection pour l'extraction des textures des images sont décrits à la section 4.2.

L'interaction de l'utilisateur avec le système de reconstruction réside principalement dans la modélisation de primitives géométriques bidimensionnelles sur les images et la spécification de correspondances et de contraintes entre ces dernières. L'accumulation de ces structures dans chaque image nous permet de résoudre de façon plus précise des systèmes d'équations nous permettant de raffiner les matrices de projection des images et les modèles tridimensionnels reconstruits.

La première étape de reconstruction consiste en la spécification de six points dans une image. L'utilisateur doit fournir les coordonnées tridimensionnelles de ces points ce qui nous permet de calculer la matrice de projection pour cette image. L'annexe C nous présente diverses propriétés utiles de cette matrice de projection pour trouver, entre autres, la position de la caméra d'une image. Une fois que les matrices de projection ont été calculées pour deux images, la géométrie se retrouvant à la fois dans les deux images

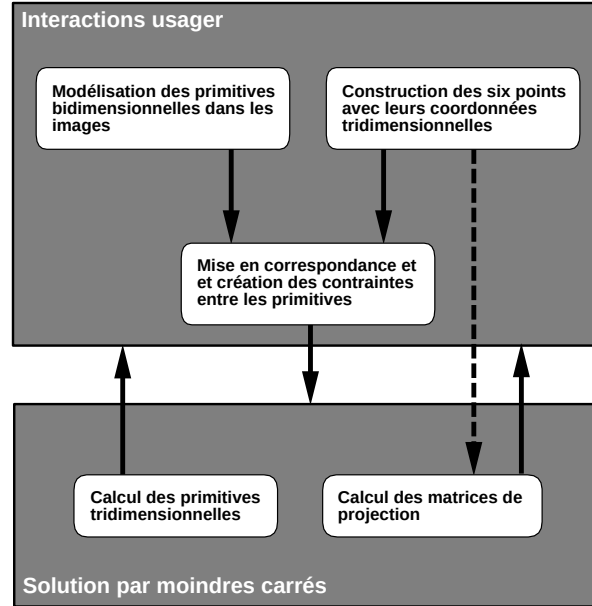


FIG. 3.2: Schéma des étapes de reconstruction géométrique.

peut être reconstruite à partir des correspondances et des contraintes. Comme expliqué dans l'annexe C, une ligne sur une image représente un plan dans l'espace (voir les équations C.2 et C.3). Pour chaque point dans une image on a une ligne horizontale et une ligne verticale donc deux plans s'intersectant en une droite dans l'espace. Si deux points dans deux images sont mis en correspondance, alors l'intersection des quatre plans ou des deux droites dans l'espace provenant de ces deux points nous donnera le point tridimensionnel correspondant aux deux points bidimensionnels projetés dans les images. L'intersection de tous les plans peut ne pas correspondre à un point tridimensionnel exacte car il s'agit d'un système surdéterminé d'équations (quatre équations et trois inconnues). La solution du système d'équations linéaires formé par ces plans par moindres carrés représente le point d'intersection qui minimise l'erreur globale, c'est-à-dire la "moyenne" des intersections de tous les plans. Les contraintes de parallélisme, de perpendicularité et de planarité se traduisent dans un système d'équations linéaires qui s'intègre de la même façon que les correspondances. La figure 3.2 nous montre un schéma des différentes étapes de reconstruction géométrique de notre système.

Des itérations multiples dans ce processus font converger les matrices de projection et les modèles vers des solutions de plus en plus stables et précises. La simplicité du processus se reflète dans l'efficacité des interactions de l'utilisateur, dans la grande rapidité

du moteur de calcul ainsi que dans la facilité avec laquelle un tel système peut être implanté. L'annexe A discute de l'implantation particulière de ce processus de reconstruction géométrique dans le cadre de ce travail et discute des extensions possibles au système.

## Chapitre 4

# Reconstruction des textures

Avant l'utilisation des textures dans la définition des modèles synthétiques, ces derniers comportaient des lacunes notables au niveau de la complexité. Cette évaluation de la complexité d'un modèle est toujours mise en relation avec le rendu du modèle. Plus le modèle est complexe, plus il contient d'informations, et plus les rendus de ce dernier seront réalistes pour autant que les techniques de rendu soient bien adaptées aux données qui définissent le modèle. Habituellement, ce sont les techniques de rendu qui définissent les informations qui entrent en jeu dans la définition des modèles. Conséquemment, il est rare de retrouver une représentation qui ne s'adapte pas parfaitement à au moins une technique de rendu.

L'apport des textures se situe à la fois sur l'aspect rendu que l'aspect complexité. D'une part, une texture est un moyen très simple pour complexifier l'apparence d'une surface, pour représenter de façon discrète ce qui se passe à un niveau qu'on pourrait appeler microscopique [BN76][CG85]. D'autre part, les textures sont très bien adaptées à plusieurs techniques de rendu très différentes et constituent plus qu'une simple structure de stockage des données. Bien que le principe d'apposition des textures sur une surface (*texture mapping*) reste toujours plus ou moins le même, les représentations des textures ainsi que les techniques de rendu les utilisant sont tellement diversifiées qu'on peut certainement évoquer une texture comme un concept beaucoup plus fondamental que l'idée plus ou moins simple et directe qu'on s'en faisait, c'est-à-dire une simple technique de *shading* [Hec89]. Dans notre cas, nous utiliserons la représentation la plus standard et la plus courante des textures, soit un tableau de couleurs ou une image. Cette représentation évoque bien la simplicité d'utilisation des textures pour

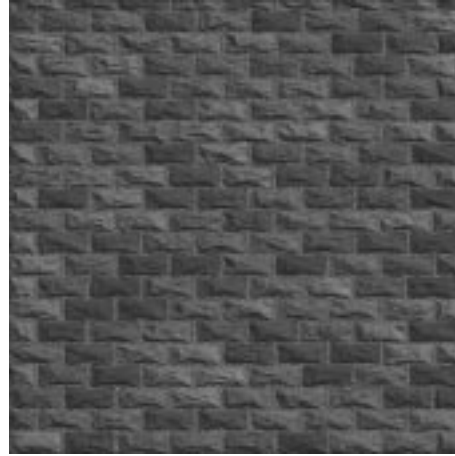


FIG. 4.1: Exemple de texture procédurale [Res96]

faire des rendus réalistes efficacement. Les techniques de rendu tirant profit de modules matériels, présentées par exemple par Hanrahan et Lawson [HL90], montrent le caractère prédominant de l'utilisation des textures en infographie.

L'obtention des textures, tout comme la modélisation géométrique, pour un modèle synthétique donné se fait dans la plupart des cas avec l'intervention indispensable d'un usager. Les techniques standards d'utilisation des textures de couleurs qui consistent à les appliquer sur une surface géométrique impliquent qu'on puisse utiliser une très grande variété d'images comme textures qui sont créées de plusieurs façons. Entre autres, un domaine de l'infographie s'intéresse à la génération procédurale des textures [Pea85][Per85] (voir la figure 4.1). Par contre, même avec l'aide de l'usager, ces techniques procédurales réussissent difficilement à simuler de manière réaliste les textures de nature réelle qu'on veut obtenir (la couverture d'un livre ou les touches d'un clavier par exemple).

Un autre moyen très utilisé pour obtenir des textures est l'utilisation des photographies digitalisées (voir la figure 4.2). Habituellement, quand on modélise des scènes, on veut apposer des textures sur les objets pour que ces derniers aient l'apparence la plus réelle possible et une photographie de la texture d'un objet est un moyen rapide et simple d'obtenir une texture extrêmement réaliste. Les photographies servant de textures sont prises à des angles, des distances et sous des éclairages particuliers afin de limiter les effets de distorsions dues aux transformations de perspectives et de réduire l'influence de l'illumination. Ainsi, on peut dire que les photographies de textures sont

“spécialisées” car elles peuvent rarement servir à d’autres applications que d’être des textures.

Les photographies spécialisées, dédiées à servir de textures, peuvent comporter des défauts de sources intrinsèques qu’un usager doit corriger manuellement. La section 4.2 présente les différents problèmes inhérents aux photographies : bruit, flou, discrétisation et normalisation des couleurs, etc. À titre d’exemple, dans la figure 4.2, on peut facilement observer la résolution finie et le bruit contenue dans les textures rapprochées à droite alors que tout paraît bien dans la scène de gauche qui est d’un point de vue plus éloigné. De plus, ces photographies contiennent aussi des défauts de sources extrinsèques, c’est-à-dire provenant de l’illumination (réflexions spéculaires), d’autres éléments de la scène (ombres, surfaces à angle rasant, occlusions), etc. Ces derniers problèmes sont très difficiles à corriger parce que nous n’avons aucune information sur la configuration scénique sous-jacente à la photographie. Par exemple, il est très difficile de produire une texture d’une façade d’une maison qui ne comporte aucune illumination particulière et qui soit exempte des effets de perspective. Dans notre cas, nous sommes en mesure d’utiliser la scène reconstruite à partir des images pour pallier aux problèmes de perspective et en partie aux problèmes d’illumination dans les images. L’utilisation de la scène pour extraire et corriger les textures constitue certainement un avantage marqué dans l’efficacité de modélisation de scènes complexes et réalistes. Dans la conclusion (voir le chapitre 5) nous apporterons des nuances quant aux avantages directs de notre technique de reconstruction des textures en comparaison des techniques standards des photographies spécialisées.

Ce travail met l’emphase sur la reconstruction et l’unification des textures provenant d’images. Il importe donc de définir le vaste domaine entourant les textures et leurs nombreuses utilisations en infographie. Les sections suivantes mettent en situation les diverses techniques reliées aux textures avec notre recherche. De plus, elles exposent le coeur de notre projet en décrivant notre approche de la reconstruction et de l’unification des textures.

## 4.1 Théorie des textures

Une texture est une structure simple pour stocker diverses informations. En fait, une texture est une image bidimensionnelle à résolutions finies qui comporte son propre

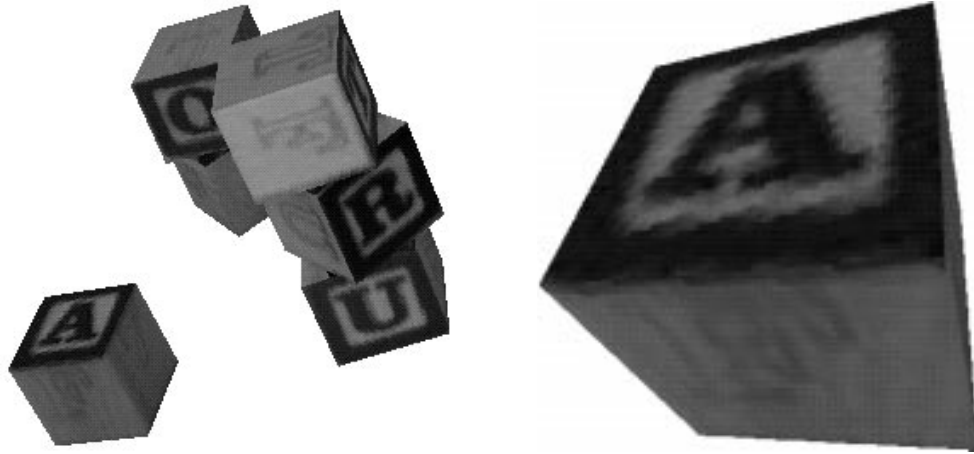


FIG. 4.2: Exemple d'une scène comportant des textures provenant de photographies

référentiel de coordonnées. Un peu comme une matrice aux dimensions finies, une texture peut être vue comme comportant un nombre fini de lignes et de colonnes. Habituellement, on dénote le référentiel de coordonnées d'une texture par  $UV$ . Chaque élément de texture a sa propre coordonnée  $(u, v)$  dans le référentiel et est nommé *texel* pour *texture element*. Son utilisation en infographie réside principalement dans sa manière de stocker les données. Lorsqu'on veut modéliser une surface plane finie, il devient vite redondant de modéliser chaque élément de surface géométriquement. Il est beaucoup plus facile d'y appliquer une image qui contiendra, à une résolution finie, les différentes données sur la surface.

L'utilisation initiale des textures dans le rendu concernait des textures de couleurs provenant d'images de la réalité [BN76] ou générées de façon procédurale [Pea85][Per85]. Cette technique d'apposition des textures sur des surfaces est toujours un moyen privilégié en infographie pour augmenter facilement la complexité des modèles et, par conséquent, le réalisme dans le rendu sans affecter la coûteuse détermination de la visibilité. Les différents problèmes d'aliassage reliés à la résolution des textures et à l'échantillonnage de ces dernières ont par la suite été partiellement solutionnés par des techniques de filtrage et de stockage des données telles les *mip-maps* [Wil83] et les *summed-area tables* [Cro84].

La technique d'apposition de textures (*texture mapping*) s'est considérablement étendue par la suite par la diversité des données pouvant être inscrites à l'intérieur d'une texture. Avec le même principe d'apposition, on peut décrire non seulement les

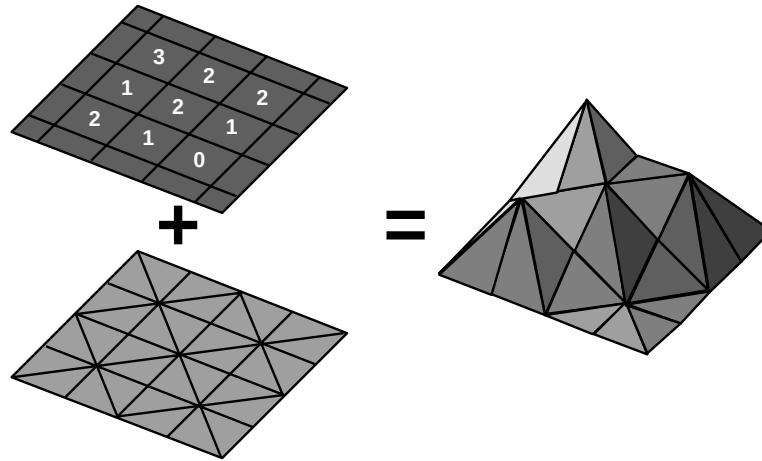


FIG. 4.3: Une texture comportant des *déplacements* ou des *hauteurs* est apposée sur une surface pour déformer sa géométrie.

couleurs sur une surface, mais aussi des déformations appliquées aux normales d'une surface. Blinn [Bli78] introduisit une méthode permettant de modifier les normales à la surface avant l'illumination de sorte que cette dernière apparaisse bosselée ou formée d'une micro-géométrie complexe comparée à la planarité et la rigidité d'une surface comportant une texture de couleurs seulement. Pour ce faire, il utilise un *bump map*, une texture qui définit une variation à appliquer à la normale dans chaque texel au lieu d'une couleur. Un déplacement ou une hauteur peut être aussi stocké dans une texture. Dans la technique introduite par Cook [Coo84], les variations de normales de la technique de Blinn sont simplement remplacées par un déplacement de la géométrie le long de la normale à la surface ce qui amplifie grandement la complexité des surfaces surtout sur les silhouettes qui sont véritablement bosselées contrairement à celles de la technique du *bump map*. Une texture comportant différentes hauteurs représentant un relief (*height fields*) peuvent aussi servir dans la modélisation et le rendu de terrains [FS80][MKM89]. La figure 4.3 nous montre un exemple de *height field* ou *displacement map* appliqué sur une surface triangularisée.

Carey et Greenberg [CG85] présentent des textures qui stockent diverses propriétés de surfaces comme des transparences au lieu des couleurs, pour calculer l'illumination et les ombres dans le rendu. Des textures comportant des transparences et différents facteurs d'opacité peuvent être utilisés pour faire le rendu de nuages par exemple [Gar85].

Nous avons vu brièvement qu'une texture, mis à part son utilisation dans le rendu

standard sous forme de couleurs apposées sur un polygone, est une structure de données suffisamment générale pour être utilisée dans une foule d'applications intéressantes. Sa manipulation facile et flexible lui confère aussi sa capacité d'action à de très bas niveaux, jusqu'à des implantations matérielles pour le rendu réaliste en temps réel. Des techniques de stockage et d'apposition en passant par les méthodes de filtrage, Heckbert [Hec86][Hec89] présente une converture détaillée de l'utilisation des textures en infographie.

## 4.2 Extraction des textures

L'extraction des données texturales est une des rares étapes de reconstruction de notre système qui n'exige pas de façon importante l'intervention de l'utilisateur. Les informations fournies par la scène reconstruite, les matrices de projection ainsi que les images sont suffisantes pour faire l'extraction autonome des textures pour chaque primitive (face polygonale) de la scène. En fait, les informations extraites des images et de la scène forment plus qu'une texture, elles sont toutes les informations que nous avons jugées pertinentes pour le calcul d'une texture finale pour une facette donnée (voir la section 4.3). D'après la diversité des phénomènes représentés dans les images, de la variété de la géométrie qui y est contenue jusqu'à la complexité des innombrables effets d'illumination, les informations pertinentes à calculer pour pouvoir produire la "meilleure" texture possible sont excessivement difficiles à déterminer. D'une part il est impossible de formaliser exactement ce qu'est une "bonne" texture et d'autre part il est très difficile d'évaluer à quel point une information particulière influence notre jugement sur la texture.

Cette section se concentre sur la présentation de notre technique d'extraction des textures (ou des informations en vue de produire des textures) des images qui ont servi à la reconstruction géométrique d'une scène. Les images de la scène à reconstruire comportent beaucoup plus d'informations sur cette dernière que sa simple description géométrique. Elles nous donnent des informations sur l'interaction de ses surfaces avec la lumière, et par conséquent sur sa micro-géométrie et ses diverses propriétés de surface, ainsi que sa couleur et plus particulièrement sa texture. La section 4.2.1 explique le procédé d'échantillonnage des couleurs dans les images utilisées dans notre système. Les techniques de filtrage impliquées dans notre méthode seront aussi présentées. La

section 4.2.2 présente les diverses informations calculées à partir des images et de la scène reconstruite, nous servant subséquemment à l'unification des texels.

#### 4.2.1 Échantillonnage et filtrage

Les primitives considérées pour l'extraction des textures sont des quadrilatères. Comme décrit dans l'implantation du système (voir la section A.2), d'autres primitives géométriques telles des triangles ou des polygones plus généraux pourraient facilement être intégrés dans le système pour faire l'extraction des textures. Toutefois, nous avons jugé suffisant d'implanter la méthode en se concentrant sur les quadrilatères puisqu'ils sont plus rapprochés de notre représentation rectangulaire de la texture et ne demandent pas de technique d'apposition (*mapping*) de texture particulière. Chaque dimension de la texture à extraire correspond à une dimension sur le polygone à quatre côtés. Ainsi la paramétrisation de la primitive pour l'échantillonnage correspond directement à la paramétrisation de l'espace texture. Une coordonnée sur le plan du quadrilatère correspond à la même coordonnée dans la texture de ce dernier. Pour une résolution de texture donnée, on parcourt la primitive dans les deux dimensions pour obtenir un point d'échantillonnage tridimensionnel (étape 1 de la figure 4.5). La précision des modèles géométriques (planarité, côtés à angle droit, etc.) a beaucoup d'incidence sur l'échantillonnage dans les images. Dans le cas des primitives non planaires en particulier, les lignes d'échantillonnage adjacentes sur la primitive n'auront pas nécessairement la même "vitesse" d'échantillonnage dans l'image ce qui peut causer de la distorsion dans la texture.

C'est pourquoi nous réduisons les erreurs en effectuant une correction bilinéaire des coordonnées d'échantillonnage tridimensionnelles. Ces corrections s'appliquent bien pour pallier au problème des angles dans les coins de la primitive. Pour ce qui est du problème de planarité de la primitive, il faudrait en plus trouver la primitive plane moyenne se trouvant entre les sommets de la primitive non plane. Le problème de planarité survenant avec une très faible erreur, les effets de distorsion de l'échantillonnage sont principalement inexistantes ou très faibles de sorte que nous ne nous sommes pas attardés spécialement à ce problème dans ce travail.

Le schéma de gauche de la figure 4.4 nous montre les échantillons non-interpolés pour une texture se trouvant sur un quadrilatère mais qui ne correspond pas à un pa-

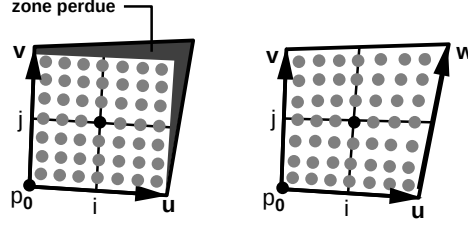


FIG. 4.4: Les points d'échantillonnage d'une texture déterminés par interpolation bilinéaire.

rallélogramme. La position tridimensionnelle des échantillons est de la forme suivante :

$$P(i, j) = p_0 + i \frac{\vec{u}}{l} + j \frac{\vec{v}}{h} \quad i \in [0, l], j \in [0, h] \quad (4.1)$$

où  $l$  et  $h$  sont la largeur et la hauteur de la texture en texels.

L'échantillonnage par interpolation bilinéaire est montré dans le schéma de droite de la figure 4.4. La position des échantillons pour ce dernier cas est déterminée de la façon suivante :

$$P(i, j) = p_0 + i \frac{\vec{du}(j)}{l} + j \frac{\vec{v}}{h} \quad i \in [0, l], j \in [0, h] \quad (4.2)$$

et

$$\vec{du}(x) = \vec{u} + x \frac{\vec{w} - \vec{v}}{h} \quad x \in [0, h]. \quad (4.3)$$

Pour chaque point  $(i, j)$  de la texture, on trouve son point d'échantillonnage  $(x, y, z)$  sur la primitive pour ensuite calculer son homologue bidimensionnel  $(s, t)$  de l'image avec la matrice de projection de l'image supportant la texture. Ainsi, nous obtenons un échantillon de couleur provenant de l'image et correspondant au point en 3D tel que vu par cette caméra. La figure 4.5 nous montre les étapes d'échantillonnage d'une couleur dans une image pour un texel donné. Dans un premier temps (étape 1), pour un texel  $(i, j)$ , on trouve le point tridimensionnel  $(x, y, z)$  correspondant sur la primitive. Par la suite à l'étape 2, on projette ce point dans l'image à l'aide de la matrice de projection de l'image pour obtenir le point d'échantillonnage  $(s, t)$ . La projection de la dernière étape s'effectue comme suit :

$$\begin{bmatrix} s \\ t \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T_0 & T_1 & T_2 & T_3 \\ T_4 & T_5 & T_6 & T_7 \\ 0 & 0 & 0 & 0 \\ T_8 & T_9 & T_{10} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (4.4)$$

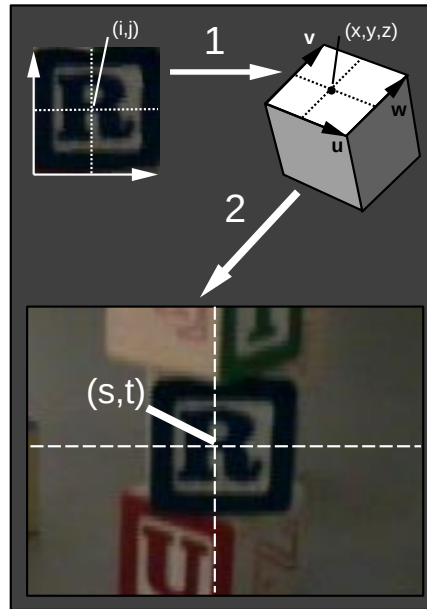


FIG. 4.5: Les différentes étapes d'échantillonnage d'une couleur dans une image.

L'erreur sur la reconstruction géométrique et la résolution de l'image sont des facteurs à considérer lorsqu'on fait l'extraction de la couleur de l'échantillon. La figure 4.6 illustre les différents types de filtrage pouvant s'appliquer à la fois au niveau des échantillons  $3D$  et au niveau des pixels de couleurs dans les images. Puisque la résolution d'une texture est discrète, chaque texel sur le quadrilatère en  $3D$  correspond lui aussi à un quadrilatère en  $3D$ . D'une part, on a la possibilité d'effectuer un sur-échantillonnage uniforme de chaque texel en  $3D$  et de pondérer les données obtenues par ces échantillons par une Gaussienne par exemple. De plus, chaque échantillon ponctuel projeté dans l'image peut recueillir les couleurs de cette dernière en les pondérant aussi selon des filtres tels que des filtres Gaussiens, pyramidaux, boîtes, etc. Cette technique bivalente de filtrage, en  $3D$  et en  $2D$ , nous donne beaucoup de contrôle sur les manières d'extraire les informations dans les images.

Habituellement, pour extraire toute l'information d'une texture d'une image et éviter l'altération inutile de la couleur nous faisons un sur-échantillonnage avec un filtre delta sur toute la primitive (voir la figure 4.7). Ce moyen d'échantillonnage est le plus simple car on va chercher directement la couleur de chaque pixel à l'aide d'un point (*point sampling*). Ce procédé de sur-échantillonnage engendre une certaine redondance dans les résultats de la texture car en moyenne, la couleur d'un même pixel

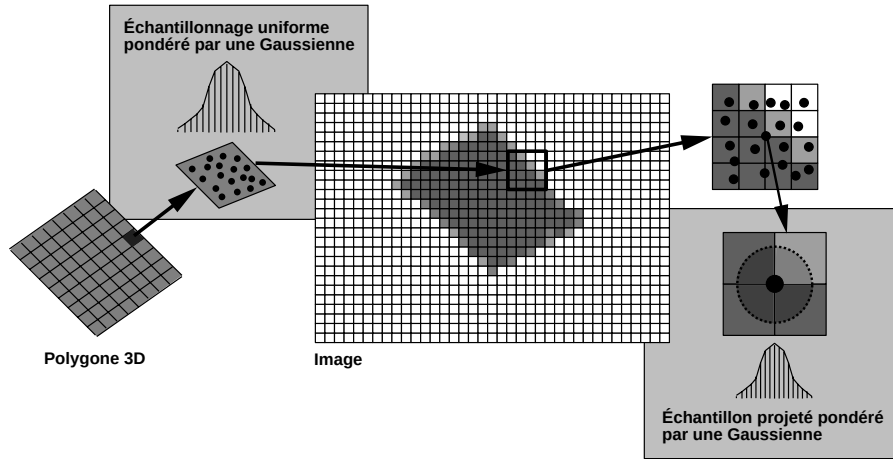
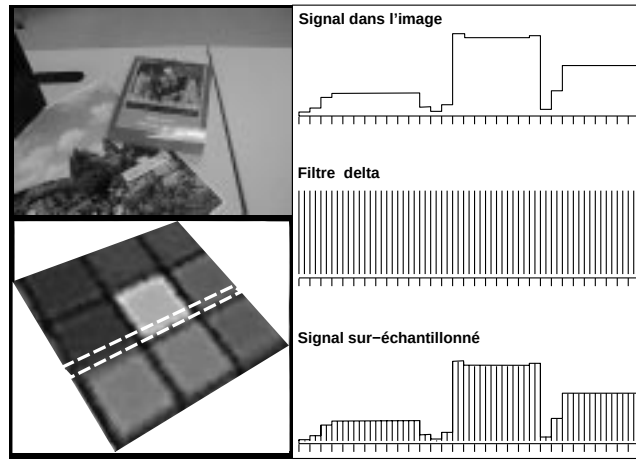
FIG. 4.6: Échantillonnage et filtrage de l'information d'une image en  $3D$  et en  $2D$ .

FIG. 4.7: Sur-échantillonnage ponctuel du signal dans une image.

se retrouve dans plusieurs texels de plus grande résolution. Grâce à la matrice de projection  $3D - 2D$ , nous éliminons directement les effets de perspective sur la texture dans chaque image.

#### 4.2.2 Construction des texels

Dans le processus de reconstruction géométrique d'une surface (voir le chapitre 3), les images contenant cette dernière contribuaient à la solution du système pour sa position et son orientation. Les sommets de la surface étaient mis en correspondance dans chacune des images concernées et nous calculions la position de ces derniers en  $3D$  à l'aide des correspondances et des contraintes fournies. Les images contribuent

de la même façon à recréer les informations de texture pour la surface. Par contre, les informations de couleurs provenant des images sont beaucoup plus complexes et ne peuvent pas être simulées par une simple projection.

Le point central de ce travail consiste à déterminer les facteurs contribuant à la couleur d'un pixel et l'intégration de ces facteurs dans une méthode d'extraction et d'analyse des textures provenant d'images. En plus de la géométrie, nous voulons obtenir les propriétés qualitatives des objets encodées sous forme de textures. De plus, il ne suffit pas de stocker toutes les informations des images pour ainsi obtenir une structure lourde et difficile à analyser. Ce genre de structure devrait contenir idéalement les images de départ ainsi que la scène reconstruite avec ces images pour que toute l'information pertinente ne soit pas perdue. Malheureusement, cette structure permettrait difficilement la réutilisation des textures dans d'autres scènes à moins de faire un traitement de généralisation comme celui que nous présentons dans ce travail.

On veut obtenir une scène standard qui contienne l'information requise pour en faire un rendu possiblement aussi réaliste que les images d'où elle est extraite. Pour ce faire, on doit "désassembler" le rendu des objets dans les images, et remonter au début, à la définition même des objets qui est indépendante de l'illumination. On doit faire un rendu inverse pour arriver à la scène de départ. L'énoncé du problème est simple en fait : à partir de photographies ou d'images d'une scène, donner la description de la scène ayant produit ces images. Dans notre cas, la description géométrique de la scène (polygones) est un problème bien cadré et donne des résultats relativement précis (voir le chapitre 3). La description texturale de la scène par contre, est un problème non résolu que nous tentons de simplifier. Une fois que les textures sont extraites des images, il faut les traiter en vue d'obtenir une texture complète sans illumination particulière (régions spéculaires et ombragées).

Une texture est "complète" si pour tout point d'échantillonnage de la primitive tridimensionnelle il existe au moins un échantillon bidimensionnel correspondant à la texture dans une image. Comme nous allons le voir, le traitement de la complétude consiste majoritairement dans une analyse géométrique des éléments provenant à la fois des images (éléments  $2D$ ) et de la scène (éléments  $3D$ ). En plus de la géométrie, l'illumination dans les images affecte tout particulièrement les données que nous traitons par nos méthodes d'extraction et d'unification des textures. Les effets d'illumination

seront traités brièvement dans ce document. En fait, la technique d'unification des textures qui sera présentée s'adresse partiellement aux problèmes d'illumination spéculaire dans les images et sur les textures.

### Occlusions

La complétude d'une texture est en grande partie affectée par l'occlusion bien qu'il soit possible qu'une partie d'une primitive puisse se trouver à l'extérieur d'une image. Lorsqu'on effectue l'échantillonnage d'une texture à partir de sa primitive tridimensionnelle, on doit tenir compte de l'information qu'on extrait. On a plusieurs points de vue différents d'une primitive et il est probable que dans certaines images, une partie de la primitive soit cachée par une autre. Dans ce cas, l'information extraite des images en cette partie doit être rejetée parce qu'elle ne fait pas partie de la texture de la primitive.

Le calcul des occlusions implique le calcul des paramètres de la caméra pour chaque image. Bien qu'une caméra comporte plusieurs paramètres, il nous suffit d'avoir la position et la direction de vue de la caméra pour déterminer la visibilité entre les primitives d'une scène projetée dans l'image. La matrice de projection obtenue par la reconstruction pour chaque image (voir la section 3.4) nous donne les informations nécessaires pour extraire la position et la direction de vue de la caméra pour chaque image. Cette position et cette orientation sont indispensables pour l'algorithme de visibilité par lancer de rayons qui nous sert à déterminer les occlusions dans les images (voir la figure 4.8). La méthode utilisée pour calculer la position et la direction de vue d'une caméra pour une image est présentée à l'annexe C. Bien qu'elle soit bien appliquée à la majorité des cas se présentant lors de l'utilisation de photographies pour la reconstruction, notre technique de détermination de la position et de la direction de vue d'une caméra est sujette à l'erreur si on est aux prises avec des photographies ou des images de points de vue orthographiques. Ces dernières images ont un point de vue qui se rapproche de l'infini ce qui nous donne bien entendu une représentation erronée dans notre système informatique. À l'annexe C, une alternative est décrite pour établir la position d'un plan de projection au lieu d'un point de vue. Il est important de mentionner que dans le cas de la détermination des occlusions, ce plan pourrait remplacer avantageusement le point de vue d'autant plus qu'il ne serait pas sujet au mauvais traitement des occlusions occasionnées par les objets se trouvant entre le plan

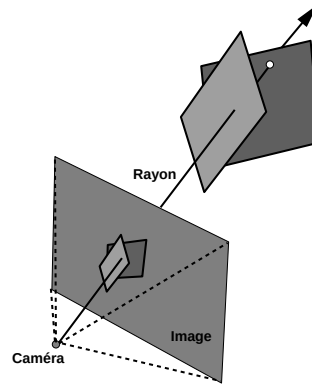


FIG. 4.8: Calcul des occlusions par lancer de rayons.

de l'image et le point de vue.

Une fois qu'on a la position et la direction de vue, on détermine la visibilité d'un échantillon tridimensionnel dans une image avec une méthode s'apparentant au lancer de rayons. On prend le segment formé de la position de la caméra et de la position de l'échantillon et on détermine si les autres primitives de la scène intersectent la droite entre ses deux points. S'il y a intersection, l'échantillon est considéré caché dans l'image et il sera marqué en occlusion de sorte qu'il ne soit pas pris en compte lors des différents calculs d'unification de texels.

La plupart du temps, les surfaces d'occlusion sont très petites en comparaison avec la surface texturale d'une primitive. Ainsi, le calcul de visibilité tridimensionnelle pour chaque échantillon est souvent inutile et ralentit considérablement le processus d'extraction des textures des images. En guise d'exemple, pour une scène comportant 110 polygones le calcul des textures sans occlusion est presque cent fois plus rapide que celui avec la détermination des occlusions. Il faut préciser toutefois que ce facteur d'efficacité dépend du nombre de polygones dans la scène, du nombre d'images où ils projettent et surtout de l'implantation vorace de l'algorithme qui ne profite d'aucune accélération provenant de la hiérarchisation de l'espace ou de l'exploitation de la cohérence entre les échantillons adjacents.

Pour accélérer le calcul, nous faisons un test simple et rapide pour déterminer s'il y a possibilité d'occlusion dans une image. Ce test consiste à déterminer pour la projection d'un échantillon donné dans une image si ce dernier est contenu dans une ou plusieurs primitives  $2D$  dans l'image. S'il y a plus d'une primitive contenant le point alors il se peut qu'il y ait occlusion en ce point et on effectue dans ce cas le calcul

complet de visibilité en  $3D$ . Sinon on n'a pas à effectuer de calcul de visibilité puisque la projection est contenue dans une seule primitive. Il faut souligner, par contre, que la véracité de ce test de visibilité en  $2D$  est directement relié au nombre de primitives modélisées dans l'image et à la précision de la projection du modèle  $3D$  sur l'image, donc à la fois sur la précision du modèle  $3D$  et sur la matrice de projection de l'image. Si la projection du modèle  $3D$  ne correspond pas exactement aux primitives  $2D$  correspondantes dans l'image, le test de visibilité peut déterminer qu'il y a une occlusion en  $2D$  alors qu'il n'y en a pas et vice-versa. Ce dernier cas où la véritable occlusion ne serait pas calculée en  $3D$  nous importe beaucoup plus que celui où l'occlusion en  $3D$  serait calculée inutilement. Pour éliminer ces cas d'échec, il suffirait de grossir les primitives  $2D$  avec lesquelles on fait le premier test de visibilité pour englober à la fois les primitives  $2D$  et la projection du modèle  $3D$  dans l'image. Ceci ferait en sorte qu'on calculerait vraiment toutes les occlusions sur une texture en s'aidant de l'accélération du test de visibilité en  $2D$  dans les images. Aussi, dépendamment de la différence entre la projection du modèle  $3D$  et des primitives  $2D$  (souvent d'au plus quelques pixels) nous aurons plus de calcul d'occlusions inutiles qui pourraient ralentir quelque peu le processus de construction des texels.

Les informations sur les texels présentées dans cette section ainsi que les couleurs extraites des images (voir la section 4.2.1) sont stockées dans une structure simple qui nous servira pour l'unification finale présentée dans la section suivante (4.3). L'implantation de cette structure et les algorithmes pour la manipuler sont exposés en détails à la section A.2. En résumé, pour une facette du modèle  $3D$ , nous stockons les informations des textures provenant de chaque image. Chaque couche de l'“hyper-texture”<sup>1</sup> correspond aux données texturales d'une image. Chaque position  $(i, j)$  dans la structure correspond à une pile de texels ou à un “hyper-texel” représentant les informations pour chaque image. C'est cette pile d'informations qu'il faut traiter pour obtenir une couleur unifiée et unique pour chaque position  $(i, j)$ .

---

<sup>1</sup> Les termes “hyper-texture” et “hyper-texels” sont utilisés par Perlin et Hoffert [PH89] pour décrire de textures procédurales tridimensionnelles. Contrairement à Perlin et Hoffert, nous utilisons ces termes pour signifier qu'un élément de l'“hyper-texture”, un “hyper-texel” en l'occurrence, contient les données texturales pour sa position provenant de chaque image.

### 4.3 Unification des textures

Le but de ce projet est de produire une texture unique pour chaque facette d'un modèle reconstruit à partir d'images. Cette utilisation des textures est typique en infographie [BN76][CG85] et elle est même intégrée dans plusieurs implantations matérielles de rendu par *z-buffer*. Une fois les données texturales extraites des images (voir la section précédente), nous sommes en possession d'un ensemble d'informations plus ou moins redondantes qu'il nous faut unifier pour obtenir une seule texture de couleurs. Cette section présente en détails notre méthode d'unification des textures.

#### 4.3.1 Transformations géométriques des textures

Les projections des primitives géométriques dans les images d'où elles proviennent ne sont pas parfaites. D'une part, il est impossible pour l'utilisateur du système de reconstruction de placer les primitives dans les images avec une précision plus grande que la résolution des images le permet. Étant donné que de nombreuses couleurs sont mélangées dans plusieurs pixels autour des sommets et des silhouettes des objets, il est difficile d'établir précisément la position des sommets dans les images. Ceci occasionne des petites erreurs dans les diverses positions des sommets des primitives dans les images. La primitive tridimensionnelle reconstruite est en quelque sorte une moyenne des positions établies dans chaque image et des contraintes 3D sur elle, et par conséquent elle reflète les petites variations et erreurs sur la position de ces points. Notre méthode de reconstruction fait en sorte de minimiser la différence entre la projection de la primitive tridimensionnelle et la primitive bidimensionnelle correspondante dans toutes les images. Mais prise individuellement, une projection dans une image donnée peut diverger sensiblement de la primitive correspondante car elle est influencée par les petites erreurs dans les autres images. Entre autres, il est important de souligner que la calibration de la matrice de projection des images peut comporter elle aussi certaines erreurs et que la simulation linéaire que nous en faisons n'est pas nécessairement appropriée. En bref, la reconstruction n'est pas parfaite pour une image prise individuellement mais elle minimise les erreurs pour toutes les images participant à la reconstruction. Conséquemment, étant donné que nous faisons l'extraction des textures à partir de la projection d'échantillons provenant de la primitive tridimensionnelle, les différences entre la projection et la primitive correspondante dans l'image sont susceptibles de se

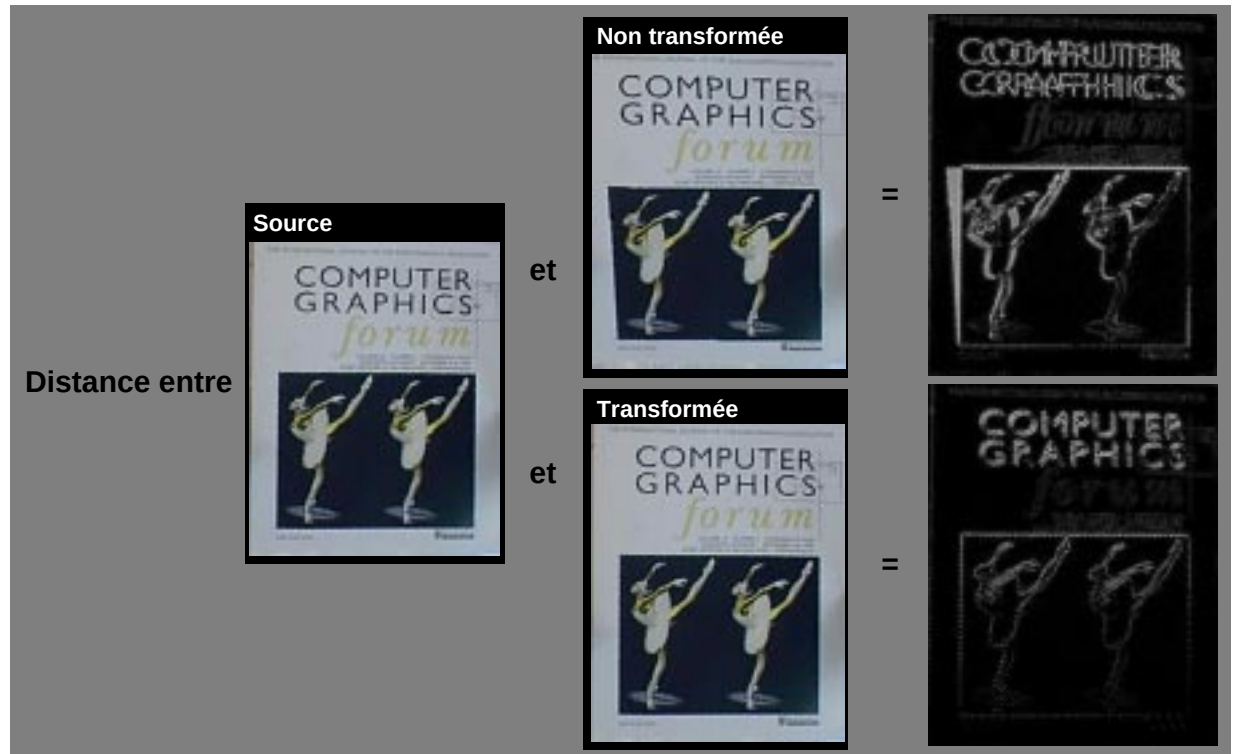
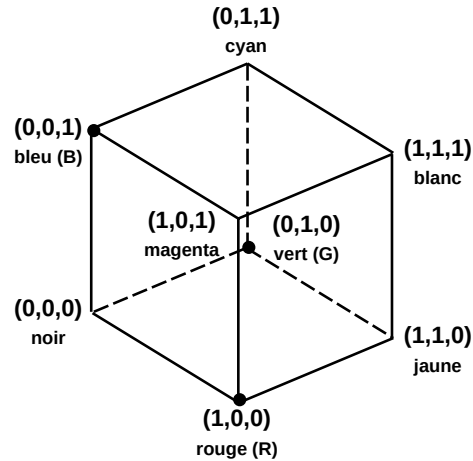


FIG. 4.9: Les images des distances entre une texture source et une texture désalignée et alignée.

refléter dans les textures extraites.

Ainsi, bien avant de traiter les détails d'unification des texels selon les diverses propriétés extraites des images et de la scène (voir la section 4.2.2), il est important de pouvoir aligner les textures de sorte qu'on puisse partiellement éliminer les effets des erreurs de projection de l'extraction. Comme nous allons le voir plus loin à la section 4.3.2, la plupart des techniques de reconstruction des propriétés de surface, de calcul de micro-déplacements sur la surface de même que certaines techniques d'unification de texels que nous avons utilisées, profitent de la différence entre les couleurs d'un même texel pour calculer différentes informations. Pour ces techniques en particulier, il est essentiel de bien aligner les textures afin de réduire les différences de couleurs causées par un décalage géométrique.

La figure 4.9 nous montre un exemple du mauvais alignement des texels causé par les différences de projection dans les images. L'intervention de l'utilisateur est requise pour établir les correspondances entre les textures extraites de chaque image. La méthode des moindres carrées (voir l'annexe B) est une fois de plus utilisée pour calculer les

FIG. 4.10: Cube représentant l'espace de couleurs *RGB*.

matrices de transformation entre les textures. Nous pouvons voir l'image des distances entre une texture source et une texture non-alignée et alignée grâce à notre matrice calculée avec système surdéterminé obtenu à partir de quatre points sur chaque texture. Nous constatons facilement que l'alignement des textures réduit beaucoup la distance entre les couleurs des texels. Bien que cette transformation ne fait pas la corrélation directe entre tous les texels des textures, elle rapproche suffisamment les textures pour réduire considérablement les erreurs occasionnées par les trop grandes différences dans notre technique d'unification des textures. Il faut ajouter que dans notre cas, nous avons une situation idéale de textures suffisamment rapprochées pour pouvoir effectuer un alignement automatique basé sur tous les texels qui serait de précision supérieure à celle obtenue par notre technique manuelle. Une transformation générale qui associerait directement tous les texels d'une texture avec ceux d'une autre texture serait beaucoup plus complexe qu'une simple transformation linéaire comme nous avons utilisée. De plus, on voudrait idéalement que ce genre de transformation considère les différences dues directement à l'illumination des objets par exemple, mais étant donné que les informations proviennent de sources différentes dans les textures, une transformation géométrique parfaite, même non-linéaire, est impossible. Cependant, la projection du modèle 3D étant d'une grande précision sur les images [Fra98], le désalignement entre les textures est souvent très faible et on observe conséquemment que la correction linéaire de ces dernières n'est parfois même pas nécessaire et est très souvent satisfaisante.

Le calcul de la distance entre les couleurs texturales s'est effectué en considérant les

couleurs comme des vecteurs tridimensionnels dans l'espace *RGB* (voir la figure 4.10) :

$$d(u, v) = |\vec{t}_1(u, v) - \vec{t}_2(u, v)| \quad (4.5)$$

où les  $t_i$  sont des textures telles que

$$\vec{t}_i(u, v) = [r_{uv}, g_{uv}, b_{uv}] \quad r, g, b \in [0, 1]. \quad (4.6)$$

Les valeurs de distances ont été reformulées dans l'intervalle  $[0, 255]$  pour former les images de distances de la figure 4.9 où le noir représente une distance nulle.

### 4.3.2 Différences de couleurs

Cette section expose les différents problèmes reliés au traitement de l'information extraite des images et de la scène reconstruite. Une fois que l'information texturale a été extraite et réalignée, on est en mesure de s'attendre à ce qu'une bonne partie des données texturales soient suffisamment bien alignées pour qu'on puisse calculer de façon plus ou moins fiable une couleur unique pour chaque texel. Bien que l'alignement des données texturales peut améliorer considérablement la solution finale d'une texture, il reste que plusieurs autres problèmes dus à l'illumination dans la scène, les propriétés de surface et la qualité des images utilisées compliquent les calculs d'unification des données jusqu'à les rendre parfois complètement inappropriés.

### Illumination spéculaire

Nous savons que les images avec lesquelles nous reconstruisons une scène sont le résultat de nombreux phénomènes plus ou moins difficiles à simuler. De ces phénomènes, plusieurs sont reliés à l'illumination dans la scène et ont beaucoup d'influence sur l'apparence des couleurs dans les images. Dans le cas des méthodes automatiques de reconstruction (voir la section 3.2), ces phénomènes peuvent hypothéquer considérablement la validité des modèles reconstruits parce qu'ils affectent directement l'apparence de la géométrie dans les images. Pour ce qui est de notre méthode de reconstruction de scènes à partir d'images, l'intervention omniprésente de l'utilisateur dans le processus d'établissement des correspondances et de la reconstruction géométrique en général nous permet d'éliminer presque complètement les erreurs pouvant être causées par les phénomènes lumineux. Cependant, dans le cas de l'extraction des textures et plus particulièrement de l'unification des données texturales, les effets d'illumination dans la

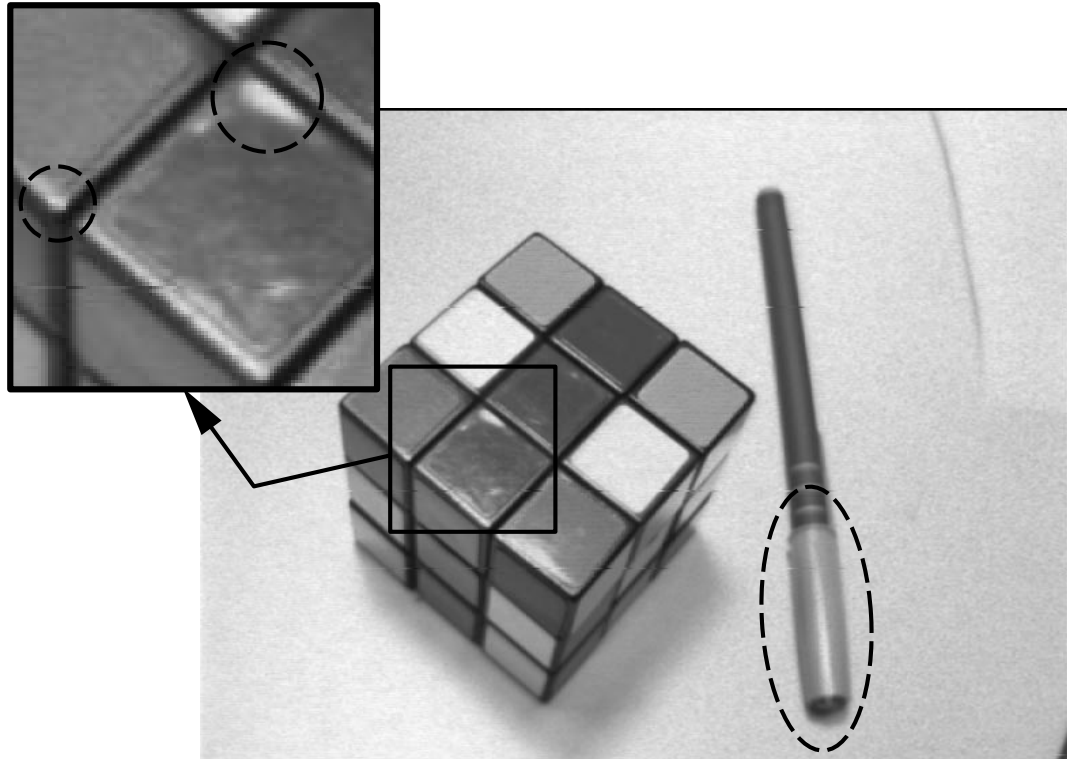


FIG. 4.11: Exemples de réflexions spéculaires sur une surface diélectrique (plastique).

scène et subséquemment dans les images sont d'une grande incidence sur l'apparence des textures. Mises à part quelques subtilités géométriques tels les micro-reliefs (voir la section 4.3.2), les couleurs dans les images sont principalement dues à la lumière et à son interaction avec les divers éléments de la scène.

Comme il sera expliqué plus loin à la section 4.3.2, les couleurs dans les images sont corrigées et transformées par les divers procédés qui les ont engendrées ce qui fait en sorte que les réflexions spéculaires sont plus facilement identifiables par le contraste qu'elles forment avec les couleurs des pixels adjacents. En présence de plus qu'une image d'une même primitive donc d'une même texture, elles sont d'autant plus facilement détectables par la différence de couleurs avec les couleurs des autres images sans réflexion au même endroit sur la texture. Malgré cela, il est intéressant de constater que les réflexions spéculaires dans une image sont souvent concentrées dans les couleurs les plus intenses de l'image. La figure 4.12 nous montre l'histogramme des tons de gris ( $[0, 255]$ ) de l'image contenue dans la figure 4.11. Les valeurs qui correspondent aux réflexions spéculaires dans l'image (indiquées par des cercles pointillés) sont exclusivement contenues dans l'intervalle  $[200, 255]$  indiqué sur l'histogramme. Une ver-

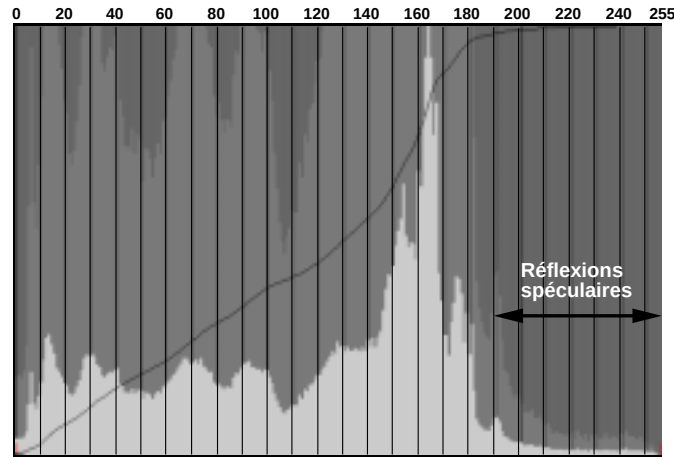


FIG. 4.12: Histogramme des couleurs de l'image contenue dans la figure 4.11

sion modifiée de l'algorithme d'unification par écarts types, présenté à la section A.2.3, intègre la contribution de ce principe simple de discernement de la couleur des réflexions spéculaires en permettant à l'utilisateur de spécifier un seuil d'acceptation sur la distance ou l'écart type d'une couleur par rapport aux autres. Des outils de visualisation des histogrammes des textures et de manipulation des seuils d'acceptation pour l'unification des texels pourraient facilement être insérés dans le système de ce travail pour faciliter la tâche de l'utilisateur.

La réflexion spéculaire sur une surface est un des effets d'illumination qui nous importe plus spécialement car elle crée souvent une zone circulaire ou plus précisément ovoïde où la radiance est plus élevée qu'à d'autres endroits sur la surface. Cette grande variation dans l'illumination d'une surface est notable dans les images car elle sature souvent la valeur de la couleur dans la région de la réflexion spéculaire de sorte que sur une surface diélectrique (plastique), on aperçoit une zone beaucoup plus brillante qui a la couleur de la lumière. Dans la majorité des cas, cette zone de réflexion spéculaire est très visible car les surfaces comportent diverses ombres et textures de couleurs variées, ce qui cause un contraste évident avec la zone spéculaire. La figure 4.11 nous montre un exemple réel d'interaction lumineuse avec une surface causant des réflexions spéculaires.

Dans ce travail, nous considérons les surfaces diélectriques, c'est-à-dire des surfaces dont les réflexions spéculaires sont plus ou moins de la même couleur que la lumière émise. Les surfaces métalliques ont des propriétés de réflexion différentes sur lesquelles nous ne nous attarderons pas. L'illumination des surfaces est de deux types : l'illumi-

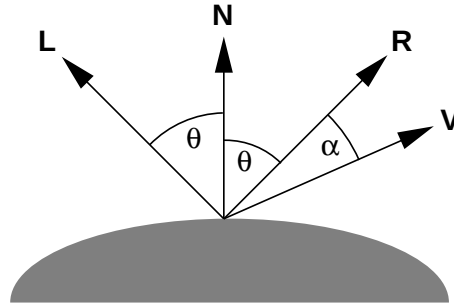


FIG. 4.13: Les différentes informations impliquées dans le calcul des réflexions spéculaires.

nation spéculaire et l'illumination diffuse. La lumière frappant une surface et réfléchié également en intensité dans toutes les directions est considérée comme l'illumination diffuse. Les reflets intenses de lumière sur la surface sont considérés comme l'illumination spéculaire d'une surface. Pour ce dernier cas, la lumière réfléchié est caractérisée par le schéma présenté à la figure 4.13.

Comme on l'a fréquemment observé dans la réalité, les réflexions spéculaires sur une surface varient selon le point de vue avec lequel on la regarde. Sur le schéma,  $\vec{N}$  représente la normale à la surface,  $\vec{L}$  est la direction d'où origine la lumière incidente,  $\vec{R}$  la direction de la réflexion parfaite ou miroir et  $\vec{V}$  la direction du point de vue. Dans le cas d'une surface miroir, la réflexion spéculaire peut être aperçue seulement quand l'angle  $\alpha$  est 0, c'est-à-dire que  $\vec{V}$  correspond à  $\vec{R}$ . Pour les autres surfaces qui ne réfléchissent pas parfaitement comme un miroir, les réflexions spéculaires seront plus ou moins atténuées et étendues. Phong [Pho75] utilise un modèle d'illumination simple pour simuler ce genre de surfaces plus rugueuses. Il utilise le terme  $\cos^n \alpha$  pour approximer les atténuations plus ou moins rapides des réflexions spéculaires sur une surface. Le facteur de rugosité de la surface est l'exposant  $n$ . Ce dernier varie de 1 (pour une surface très rugueuse) à l'infini (pour un miroir parfait).

Heureusement, comme nous l'avons vu plus haut, l'apparition de réflexions spéculaires sur une surface dépend du point de vue avec lequel on observe cette dernière. Ainsi, on peut espérer pouvoir discerner entre les couleurs qui sont le résultat d'une réflexion spéculaire et ceux qui ne le sont pas, étant donné que nous avons des images de différents points de vue. Pour des points de vue assez différents d'une même surface, on peut s'attendre à éliminer la majorité des effets de réflexions spéculaires puisque

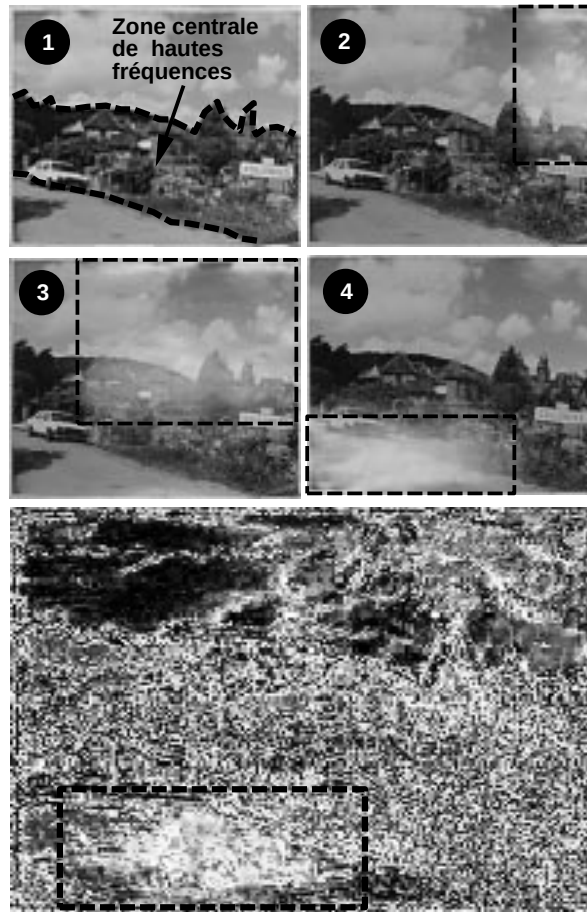


FIG. 4.14: Image de disparités (au bas) obtenue par l'unification de quatre textures comportant une zone de réflexion spéculaire.

ces dernières sont susceptibles de se retrouver dans des endroits différents dans chaque image. Le fait qu'on puisse discerner les réflexions spéculaires dans les images dépend directement de la rugosité de la surface. Plus une surface est rugueuse, plus ses réflexions seront étendues et pâles. Elles seront plus difficilement identifiables contrairement à celles d'une surface dont la réflectivité s'approche de celle d'un miroir. La figure 4.14 nous montre des textures d'une même primitive dont certaines comportent des réflexions spéculaires évidentes. On peut voir aussi une image des discontinuités dans les couleurs des texels extraits.

Pour un texel donné, plus une couleur correspondant à une couche (ou à une image en particulier) s'éloigne de la moyenne des autres couleurs, plus sa contribution est considérée comme négligeable dans la couleur finale du texel. Les plus faibles contributions sont marquées en blanc sur l'image de disparités dans la figure 4.14. Nous

remarquons que les textures 2 et 3 dans la figure 4.14 comportent une zone de réflexion spéculaire moins intense et plus étendue que celle contenue dans l'image 4. L'image de disparités représente les écarts maximum à chaque texel pour les textures données. Ces disparités entre les couleurs des texels furent calculées avec la formule suivante :

$$disp(u, v) = \max(|\vec{t}_i(u, v) - \vec{t}_{moy}(u, v)|) \quad i \in [1, n] \quad (4.7)$$

où  $(u, v)$  est la position des texels à unifier,  $n$  est le nombre de textures impliquées dans l'unification<sup>2</sup>,  $\vec{t}_i(u, v)$  est la couleur de la texture  $t_i$  à la position  $(u, v)$  (voir l'équation 4.6) et  $\vec{t}_{moy}(u, v)$  est la moyenne des couleurs à la position  $(u, v)$  et est calculée de la façon suivante :

$$\vec{t}_{moy}(u, v) = \frac{1}{n} \sum_{1 \leq j \leq n} \vec{t}_j(u, v). \quad (4.8)$$

L'image de disparités nous montre bien les écarts types plus accrus dans la région correspondant à la zone spéculaire de l'image 4. La région centrale de l'image est affectée par la grande quantité d'informations qui correspondent plus ou moins dans chaque texture causant du bruit dans cette dernière. Le nuage situé dans le coin supérieur gauche dans les textures n'est presque pas affecté par les réflexions spéculaires et conserve donc sa couleur originale sans illumination spéculaire dans toutes les textures. Nous remarquons que les écarts types dans la région du nuage dans l'image de disparités sont relativement faibles en comparaison aux autres régions qui sont couvertes par les diverses zones spéculaires des textures.

La figure 4.15 nous montre le résultat de l'unification des textures de la figure 4.14 à l'aide de notre algorithme présenté à la section A.2. Puisque les couleurs qui proviennent de réflexions spéculaires sont faciles à identifier parmi les couleurs d'un même texel, nous avons amoindri leur influence sur la couleur finale en fonction de leur écart avec la moyenne des autres couleurs. Nous remarquons que les couleurs comportant une réflexion spéculaire dans une texture ont été supplantées par les couleurs des autres qui n'en avaient pas. On voit aussi le résultat obtenu lorsqu'on fait seulement la moyenne des texels, indifféremment de leur écart type. Nous constatons que la texture résultante (image de droite dans la figure 4.15) comporte moins de contrastes car elle intègre les couleurs provenant de réflexions spéculaires avec le même poids que les autres couleurs. Un des grands avantages de cette méthode d'unification est qu'il n'est

---

<sup>2</sup>  $n = 4$  dans l'exemple de la figure 4.14.

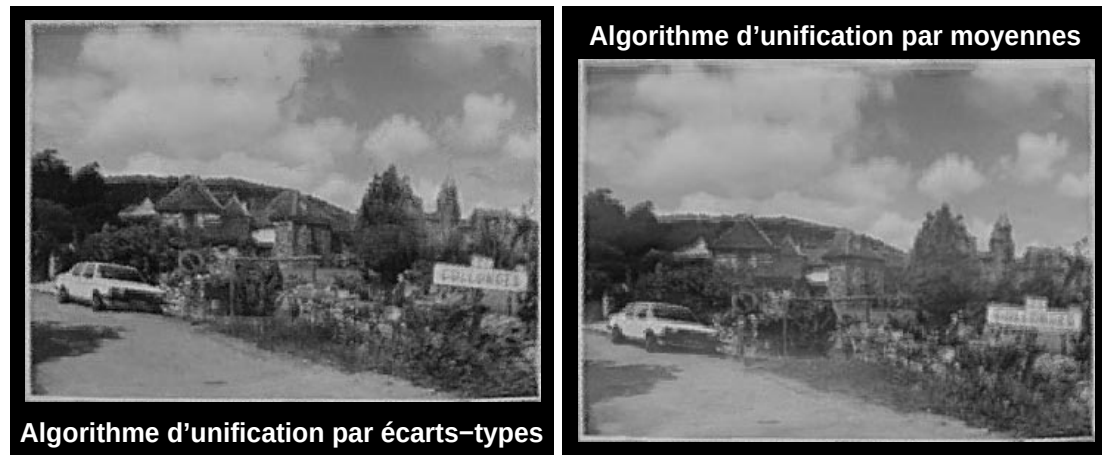


FIG. 4.15: Résultats des algorithmes d'unification par écarts types et par moyennes.

pas nécessaire de modéliser l'illumination de la scène, ce qui peut être très complexe et coûteux, pour réussir à éliminer sensiblement tous les effets spéculaires notables sur les textures. Cependant, cette méthode est très sensible au nombre de couches ou d'images impliquées dans la reconstruction d'une texture. Puisque cette méthode est basée sur l'écart à la moyenne des couleurs, il doit y avoir beaucoup moins de couleurs provenant de l'illumination spéculaire qu'il y a d'autres couleurs. De plus, non seulement le nombre de couches mais la position des réflexions spéculaires peut facilement influencer les résultats de l'algorithme. Deux réflexions spéculaires se chevauchant plus ou moins dans un texel auront nécessairement plus de poids sur le résultat final.

Bien que dans notre travail, les réflexions spéculaires dans les textures sont indésirables, il reste que certaines techniques de reconstruction des propriétés de surface à partir d'images profitent pertinemment de ces dernières. Un domaine de recherche, plus rapproché de la vision informatisée se nomme *color constancy* et consiste à reconstruire la couleur perçue d'une surface en éliminant la contribution de la couleur de la lumière et de l'illumination en général. Klinker et al. [KSK88], Funt et al. [HFD90] et Sato et al. [SWI97] nous donnent des techniques simples pour estimer les coefficients diffus et spéculaire d'une surface en fonction d'un histogramme des couleurs se trouvant dans les images et d'un modèle d'illumination particulier. Habituellement, les modèles d'illumination utilisés dans ces techniques s'appliquent à des surfaces diélectriques, c'est-à-dire que les réflexions spéculaires sur la surface sont de la couleur de la lumière.

Les techniques impliquées dans le domaine du *color constancy* sont relativement

stables pourvu que les images utilisées soient de bonne qualité et que les données soient relativement invariables entre elles. En fait, ces techniques fonctionnent presque exclusivement sur des surfaces d'une seule couleur, donc sans texture complexe qui comporte une variété de couleurs. De plus, comme nous allons voir dans la section 4.3.2, les images que nous utilisons pour la reconstruction peuvent provenir de sources variées qui comportent plusieurs problèmes. Cependant, nous ne voulons pas intégrer d'équipements sophistiqués qui pourraient amputer notre technique d'une partie de sa flexibilité et de sa généralité au profit d'une reconstruction plus rapide. En fait, soit que ces techniques de vision impliquent des situations trop simples où les conditions d'illumination de la scène sont idéales et partiellement bien modélisées, soit qu'elles requièrent la manipulation d'équipements de mesures spécialisés pour extraire des informations qui sont très difficiles à calculer à partir de simples images.

### Micro-reliefs

Nous avons vu qu'une texture est un moyen simple et versatile pour encoder une grande variété de propriétés sur une surface. Sa contribution au niveau du réalisme d'une surface est d'autant plus importante dans le contexte de la reconstruction à partir d'images puisque d'une part on obtient directement les textures des images et d'autre part, la reconstruction d'objets complexes étant difficile, nous sommes habituellement contraints de construire des surfaces polygonales qui utilisent pertinemment la complexité apparente des textures pour augmenter leur réalisme. Entre autres, établir les correspondances pour une surface courbe dans plusieurs images par exemple est une tâche compliquée puisqu'il est souvent difficile d'établir des points de repère précis d'une image à l'autre.

Cette limite sur la complexité des modèles causée par une modélisation simple avec des surfaces polygonales n'est pas seulement due à la simplicité de topologie de ces dernières mais aussi au faible nombre de celles-ci pouvant être disposées précisément dans les images par l'utilisateur. Une scène peut comporter des objets de très petites dimensions qui seront difficiles à reconstruire dans certaines images. D'autres objets seront formés d'une multitude de facettes et causeront les mêmes problèmes. Dans ces cas, une texture de couleurs peut difficilement simuler le comportement de la surface de façon indépendante du point de vue. Les petits objets se trouvant sur une surface

ou la forme subtile de cette dernière sont souvent impossibles à réduire dans une seule texture de couleurs. Comme présentées à la section 4.1, les textures de déplacements de points sur une surface (*height field* ou *displacement map*) peuvent être utilisées pour simuler un comportement géométrique sur la surface. La géométrie d'une surface trop complexe (en terme du nombre de facettes et de leur taille) peut souvent être reconstruite automatiquement à partir des images et stockée dans ce genre de textures.

La plupart des techniques de reconstruction du micro-relief des surfaces à partir d'images fonctionnent sur le même principe de comparaison des couleurs dans les images. En général, pour un polygone 3D donné, on trouve ses micro-reliefs en échantillonnant la texture de la primitive dans chaque image (suivant une méthode similaire à notre méthode d'extraction des textures) et en évaluant les différences entre les couleurs correspondantes. Si les couleurs d'un même échantillon sont similaires à un seuil d'erreur près, alors on fait l'hypothèse que l'échantillon sur la surface est à la bonne hauteur. Dans le cas contraire, on déplace l'échantillon légèrement le long de la normale du polygone et on fait la lecture dans les images jusqu'à ce qu'on obtienne des couleurs similaires pour chaque échantillon. Des techniques plus sophistiquées dont celle présentée par Scheuing et Niemann [SN86] conservent tous les résultats intermédiaires des différences entre les couleurs en leur attribuant un poids pour évaluer une surface optimale moyenne passant par le nuage des hauteurs obtenues.

### Qualité des images

Les sources de données primordiales de notre système de reconstruction sont les images. Elles comportent toutes les informations indispensables à l'établissement des structures géométriques et des textures appliquées sur ces dernières. Bien que son influence sur la reconstruction géométrique soit moins marquée dans notre cas que dans celui des méthodes de reconstruction plus automatiques (voir la section 3.2), la qualité des images est certainement un aspect dont l'analyse est d'une extrême importance quand on considère l'extraction et l'unification des textures provenant des images. Les informations texturales sont directement issues des images et leur qualité est par conséquent fortement responsable de l'exactitude de la simulation des divers phénomènes qui s'y retrouvent.

L'évaluation de la qualité des images, telle que nous l'entendons, consiste en plu-

sieurs éléments d'analyse des images qui changent l'apparence de ces dernières et faussent notre perception. Plus une image comporte de ces phénomènes indésirables, plus sa qualité sera considérée inférieure. Bien entendu, l'influence de ces phénomènes sur la validité de nos résultats varie énormément, la rendant difficile à analyser dans le contexte de ce travail. Toutefois, nous avons déduit que les phénomènes qui causent des différences entre les images, surtout sur les couleurs, sont le plus souvent ceux qui provoquent le plus d'ennuis aux techniques utilisant la cohérence entre les images comme la nôtre. Entre autres, les techniques de reconstruction de micro-reliefs présentées plus haut (la section 4.3.2) sont principalement basées sur les différences de couleurs entre les images et par conséquent, elles sont excessivement sensibles aux effets de la multitude des phénomènes pouvant affecter les couleurs. En fait, dans la quête du réalisme, nous avons besoin de beaucoup d'informations. Par contre, plus il y a d'informations, plus les chances sont minces qu'elles puissent bien s'accorder car elles dépendent de nombreux paramètres qui changent constamment, surtout d'une image à l'autre. La richesse des informations contenues dans les images comporte aussi sa part de problèmes parce que ces informations sont difficiles à corréler si elles sont moins éloignées.

Certaines images comportent du bruit ou incluent des régions de très hautes fréquences d'informations. Il est très difficile d'unifier ces régions car elles correspondent très rarement entre elles, même si elles donnent une impression visuelle semblable. La bande centrale contenue dans les textures de la figure 4.14 contient beaucoup d'informations et bien qu'elle semble d'apparence constante dans chacune des textures, elle est la cause d'un grand nombre de disparités. Plusieurs techniques d'analyse d'images peuvent corriger partiellement le bruit dans une image en filtrant les données et en éliminant ainsi les hautes fréquences qu'elles identifient comme du bruit [Wol90][GW92].

### **De couleurs à radiances**

Un problème qui survient fréquemment quand on effectue une reconstruction à partir de photographies consiste en la normalisation des couleurs dans les images. Les tons des couleurs dans une photographie sont normalement transformés en fonction d'une zone de focus ou de la région la plus intense dans l'image. Puisque la présence et les intensités de telles régions varient d'un point de vue à un autre, une même région dans deux photographies apparaîtra de couleurs différentes à cause de la normalisation

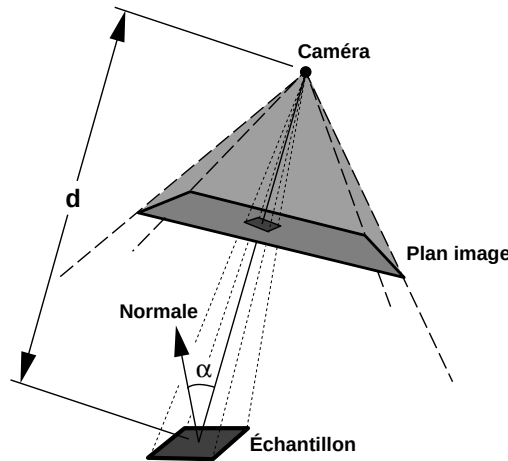


FIG. 4.16: La distance  $d$  et l'angle de vue  $\alpha$  d'un échantillon et d'une caméra.

des tons (*tone mapping*) dans les images. Debevec et Malik [DM97] ont présenté une technique pour calculer une image d'intervalles de hautes intensités nous permettant d'extraire les radiances dans la scène plutôt que les couleurs *RGB*. L'utilisation de ces images de radiances contourne les problèmes de troncation et de changement d'échelle des données dus à la résolution limitée des images<sup>3</sup> et de normalisation des données selon l'intensité lumineuse se retrouvant dans un point de vue d'une scène donnée. Il faut souligner toutefois que les radiances stockées par la technique présentée par Debevec et Malik [DM97] sont relatives pour un film donné. Elles dépendent donc d'un film particulier qui a une réponse absolue à la lumière. Ce dernier détail fait en sorte qu'on a des échelles de radiances absolues pour les photographies provenant d'un même film. Pour ce qui est des applications inter-images et intra-images telles l'extraction de l'illumination, la reconstruction des propriétés de surface et de textures réalistes, les images de radiances comportent des avantages incontournables en comparaison des images *RGB* traditionnelles.

Un problème semblable peut survenir dans le cas plus général où on traite des images synthétiques en plus des photographies. Des opérations de correction gamma sur une image par exemple peuvent affecter grandement la cohérence de couleurs qu'elle avait avec d'autres images provenant d'autres points de vue.

Certains systèmes de reconstruction [NB95][DTM96] effectuant une unification et

---

<sup>3</sup>Chaque canal de couleur, *R*, *G* ou *B*, a une résolution de 256 valeurs. Sa valeur est donc contenue entre 0 et 255.

une correction des textures provenant d'images utilisent la position de la caméra de chaque image pour évaluer la distance et l'angle d'un texel par rapport à l'image. La figure 4.16 nous montre la configuration des divers éléments participant à l'évaluation de l'angle et de la distance entre un point d'échantillonnage représentant un texel et une image. Il est intuitif d'établir qu'un point éloigné du plan image dont l'angle de vue s'approche de  $90^\circ$  risque d'être de mauvaise qualité. Dans le même sens, un texel est de meilleure qualité s'il est rapproché et parallèle au plan image. Cette hypothèse peut nous sembler logique mais ses contributions réelles dans l'évaluation de la qualité des texels sont très mitigées. Dans la majorité des scènes reconstruites par notre système, les textures qu'il fallait unifier étaient vues à des distances et des angles qui ne permettaient pas vraiment de discriminer entre les “bonnes” et les “mauvaises” vues; ceci étant dû aussi à la relative imprécision dans la reconstruction de la caméra (voir l'annexe C). Dans les cas où on est en mesure de vraiment différencier les textures éloignées et obliques au plan image, les photographies de ces dernières peuvent comporter un meilleur focus que les images des textures considérées comme idéales.

### **Pixels frontaliers**

Les problèmes de mélanges des pixels frontaliers d'un polygone dans une image, mentionnés à la section 3.2.2, affectent non seulement la mise en correspondance automatique des images, mais aussi la qualité des textures. La discrimination des couleurs faisant parties de la texture et celles provenant de l'arrière-plan ou d'une autre primitive dans une image, est très importante pour éliminer les discontinuités de couleurs indésirables et très perceptibles sur les bordures des textures. On remarque, entre autres, que les couleurs de l'arrière-plan sont mélangées avec les silhouettes et ce de façon plus ou moins importante dépendant de la distance du point de vue, du focus et de la résolution de l'image. Un problème similaire survient lorsque les textures de primitives qui partagent un même côté sont extraites d'images de différents points de vue et de différentes illuminations. La frontière entre les deux primitives pourra être très contrastée et possiblement désalignée. Niem et Broszio [NB95] diminuent les effets de discontinuité en pondérant et en “moyennant” les texels frontaliers des textures des deux primitives. Cependant, cette méthode altère les données de la texture et ne constitue pas une simulation précise des phénomènes causant ces variations entre deux

facettes. De plus, sur des textures à motifs réguliers et simples, on remarque parfois que les texels de la frontière entre deux textures ne correspondent pas exactement entre eux, ils sont désalignés. Ceci est dû à la paramétrisation différente des deux primitives surtout lors d'un échantillonnage insuffisant. Si elles ne sont pas échantillonnées dans les mêmes directions, on risque d'avoir un désalignement aux frontières. Le fait qu'on modélise principalement des quadrilatères dans notre système évite la majorité des problèmes de désalignements frontaliers dans une même texture. Ces désalignements seraient plus fréquents si on devait modéliser nos quadrilatères avec des triangles seulement.

## Chapitre 5

# Conclusion

Nous avons présenté une méthode générale pour faire l'extraction et l'unification de textures en provenance d'images. Comme nous avons pu le constater, plusieurs problèmes sont inhérents à la technique d'unification et les solutions valables sont souvent contredites par de nombreuses situations, ce qui rend la technique dépendante de l'intervention de l'utilisateur. La philosophie de notre système de reconstruction étant en partie structurée autour de ce précepte d'intervention constante de l'utilisateur, le processus de reconstruction des textures pour une scène ne contrevient en rien à briser le rythme continu du procédé global de reconstruction. Dans le cas particulier de la reconstruction des textures, nous avons vu que l'intervention humaine est indispensable et très complexe, et contrairement à la reconstruction géométrique, nous entrevoyons très peu de moyens pouvant rendre la reconstruction des textures plus automatique en gardant à l'esprit que nous désirons obtenir la "meilleure" texture possible.

Si on met l'emphasis sur le réalisme des textures et du rendu, on devra concéder que le fait d'extraire nos textures des images ayant servi à la reconstruction géométrique offre quelques avantages flagrants sur l'utilisation de photographies spécialisées. Sur le plan géométrique seulement, le fait qu'on puisse corriger les textures de la distorsion de perspective supprime amplement les quelques arguments en faveur de la qualité habituelle des photographies spécialisées. De plus, la variété des informations extraites des images, engendrées par les différents points de vue, les différentes illuminations, enfin, les diverses propriétés directionnelles des objets projetés pourraient servir à calculer beaucoup d'autres propriétés des surfaces, ce qui est impossible avec les photographies spécialisées. Cependant, ces informations sont loin d'être idéales pour

créer des textures précises et réalistes. Un grand avantage de l'utilisation des images et surtout des photographies dans le rendu est qu'on les utilise directement en joignant leurs différentes données selon un point de vue arbitraire. Qu'on soit en présence d'un modèle géométrique ou non, comme dans les techniques présentées dans le chapitre 2, le réalisme atteint par l'utilisation des données provenant de photographies peut difficilement être approché par les techniques de rendu standard (lancer de rayons, radiosité) et ce, de façon aussi efficace.

Si on envisage le rendu infographique comme but ultime de la tâche de reconstruction, on peut questionner la pertinence d'extraire les textures des images et d'en faire l'unification pour produire des textures indépendantes du point de vue et de l'illumination. Les diverses opérations que nous effectuons sur les données pendant la reconstruction des textures tendent à corrompre considérablement leur réalisme initial qui ne pourra peut-être jamais être réinstauré de façon convaincante dans les rendus a posteriori de nos modèles selon ces mêmes points de vue. Cependant, il est important de souligner que pour des points de vue généraux pour lesquels nous n'avons pas de photographie, les textures unifiées peuvent satisfaire amplement plusieurs des exigences du réalisme et ce, de façon très efficace.

L'implantation d'un système de reconstruction complet, pave la voie à de nombreuses recherches. Entre autres, la reconstruction des textures en provenance d'images sera sans aucun doute comblées d'une foule d'améliorations par les travaux futurs de recherche dans ce domaine. Avec la venue du champ de recherche sur le rendu d'un monde dual, entre le réalisme et le synthétisme, appelé réalité augmentée, la reconstruction des modèles géométriques et des textures à partir d'images deviendra une plateforme de recherche importante nous amenant à confondre le réel et le virtuel. Les scènes reconstruites par notre système forment la base de toutes les techniques d'intégration d'objets synthétiques et d'objets réels [FGR93][DRB97][Deb98]. Sous le plan de l'illumination en particulier, ces techniques peuvent grandement profiter des informations pouvant être reconstruites à partir des textures dans les images.

Les découvertes faites dans le domaine du rendu pour la réalité augmentée ouvrent plusieurs directions sur la reconstruction à partir d'images. Les couleurs se trouvant dans les images sont le résultat d'une multitude de phénomènes complexes. Discriminer ces phénomènes et les cerner précisément dans les images constitue actuellement un

des buts principaux de la reconstruction à partir d'images. L'usage des images s'étant approfondi par le stockage de radiances plutôt que des classiques triplets *RGB* [DM97], la reconstruction à partir d'images s'adresse maintenant à des problèmes beaucoup plus vastes, mais aussi plus précis tels l'extraction de l'illumination, des propriétés de réflexion et plus généralement des *BRDF* [LFTG97] des images.

Comme nous l'avons vu dans les chapitres précédents, nous nous sommes attaqués à un problème excessivement complexe ; un problème difficile à formaliser parce qu'il intègre plusieurs éléments subjectifs résolus exclusivement par l'humain. Nous n'avons qu'à imaginer les problèmes soulevés par une texture séparée en deux parties dont les couleurs proviennent d'images différentes. Si une des parties est de très faible qualité en comparaison à l'autre partie, il est évidemment très compliqué de prendre une bonne décision automatiquement. Cette décision quant à la conservation des données précises ou à la normalisation des données à la plus faible qualité réside en partie dans une évaluation esthétique impossible à formaliser. Le travail accompli et les résultats obtenus reflètent faiblement les efforts de réflexion et les questionnements qui les ont engendrés. Toutefois, notre recherche, remplie de promesses mais qui fut semée d'une multitude d'embûches, a produit un système de reconstruction fonctionnel qui réussit malgré tout à créer quelques résultats enviables<sup>1</sup> aux autres systèmes de reconstruction.

Ce travail nous a permis d'étudier le vaste domaine de la reconstruction à partir d'images. Les images utilisées pour la modélisation géométrique ne sont qu'une des nombreuses applications très utiles de ces dernières. Nous avons tenté d'exposer, dans ce travail, la grande diversité des informations pouvant se trouver dans les images. Tout en se concentrant sur une utilisation bien cernée de ces dernières pour faire la reconstruction des textures, nous avons exploré une multitude d'autres avenues les utilisant. Cette étude nous a amenés à concevoir une grande fusion entre les diverses techniques de modélisation et de rendu en infographie. Les frontières entre les modes de représentation et entre les techniques de rendu s'y référant sont maintenant vagues et distorsionnées. Nous ne sommes plus en présence de sphères de recherche dont l'objet est précis et borné. Les concepts s'étendent, se chevauchent pour former un seul univers infographique où toute les connaissances s'allient éventuellement pour l'atteinte d'un

---

<sup>1</sup>Mentionnons entre autres l'efficacité de convergence de la reconstruction, la robustesse des modèles obtenus et particulièrement l'extraction et le traitement des textures comme nous l'avons expliqué tout au long de ce travail.

même but : le réalisme.

Notre système de reconstruction qui permet l'alliage du rendu standard avec certaines notions de rendu à base d'images n'est qu'un piètre exemple de l'énorme versalité des techniques de rendu hybrides de plus en plus répandues en infographie. Éventuellement, il ne nous sera plus possible de distinguer la réalité photonique d'une caméra d'une réalité virtuelle complètement synthétique. Pour le moment, nous en sommes à faire la jonction entre ces deux réalités, et bien que les résultats de notre recherche puissent paraître plus ou moins convaincants, nous ne pouvons douter de leur pertinence si ce n'est que de l'anticipation qu'ils nous offrent vis-à-vis cet avenir prometteur pour l'infographie.

## Annexe A

# Description du logiciel

Un aspect important des techniques analysées dans ce travail réside dans leur applicabilité dans différents domaines de l'infographie. Dans notre cas en particulier, on découvre que le but réel du travail se rapproche beaucoup plus d'une solution informatique que d'une solution fondamentale et théorique. Ainsi, la solution à notre problème nous suggérant plusieurs liens intrinsèques avec l'informatique, nous trouvons important de décrire les structures et les mécanismes informatiques de base du logiciel qui résultent de notre recherche. D'autres projets de nature plus mathématique impliquent des implantations informatiques à titre plutôt accessoire de façon à corroborer les résultats des techniques. Dans notre cas, le problème et les techniques utilisées pour le solutionner proviennent directement du domaine de l'infographie et de la vision appliquée, donc de l'informatique. Cette annexe décrit la structure du logiciel, du comportement global aux différents éléments détaillés de son fonctionnement. La recherche dans ce travail s'étant beaucoup concentrée sur une implantation d'un système de reconstruction, cette annexe se consacre à la description plus détaillée des aspects informatiques de ce dernier. En plus de son apport au niveau de la compréhension des éléments théoriques de notre travail, cette annexe peut être utilisée comme référence de programmation pour un logiciel de reconstruction.

Le système implémenté au cours de la recherche qui nous concerne fut écrit dans le langage *C++*. Le grand pouvoir d'expression et la diversité des bibliothèques de ce langage en fait un choix évident pour les projets en infographie. La description du système est divisée en deux parties relativement indépendantes : la partie concernant le système de reconstruction géométrique (A.1) et celle concernant le système de reconstruction des

textures (A.2). De plus, en guise de conclusion supplémentaire, la section A.3 expose quelques corrections qui pourraient être apportées au système et donne un bref aperçu de certaines idées d'extensions pouvant faire l'objet de travaux futurs.

## A.1 Système de reconstruction géométrique

Cette section donne une description globale du système de reconstruction en se concentrant sur le procédé de reconstruction géométrique. Comme nous allons le voir subséquemment à la section A.2, le procédé de reconstruction texturale est assez succinct et indépendant du système global. Le système implémenté dans ce travail est principalement voué à la reconstruction géométrique et les divers structures et mécanismes qui le constituent sont majoritairement axés en ce sens. Les classes et leurs relations sont exposées brièvement dans cette section étant donnée la faible importance qu'elles ont vis-à-vis des concepts exposés au chapitre 3 et plus particulièrement à la section 3.4.

Notre système de reconstruction est une application (*WsRekonApplication*) globalement formée d'une session (*WsRekonSession*) et d'une interface graphique (*WsGkRekonSessionViewer*) qui nous permet de manipuler cette dernière. La session renferme toutes les données et les méthodes qui constituent les fonctionnalités générales du système. L'interface graphique globale assemble les outils pour manipuler et éditer les images, les primitives géométriques et les contraintes entre ces dernières, et déclencher les divers mécanismes de reconstruction. La session contient la liste des images (*WsRekonImageStrip*) impliquées dans toutes les étapes de la reconstruction géométrique et texturale, la scène tridimensionnelle (*WsRekonScene3D*) résultant de la reconstruction et le gestionnaire des contraintes (*WsRekonConstraintManager*) du système. La session contient aussi les interfaces graphiques nous permettant de visualiser les images (*WsGkRekonImageStripViewer* et *WsGkRekonImageViewer*) et la scène tridimensionnelle (*WsGkGeomScene3DViewer*). Le gestionnaire de contraintes<sup>1</sup> conserve une liste des contraintes du système et son interface est responsable de toutes les manipulations sur celles-ci. Chaque image (*WsRekonImage*) du système contient une scène bidimensionnelle (*WsRekonScene2D*) qui stocke les différentes primitives 2D (*WsRekonPoint2D*, *WsRekonPolygon2D*) créées dans cette image. La figure A.1 illustre la distribution des classes dans la système.

---

<sup>1</sup> Voir [Fra98] pour une présentation détaillée des concepts reliés aux contraintes mentionnés ci-haut ainsi que la littérature s'y rapportant.



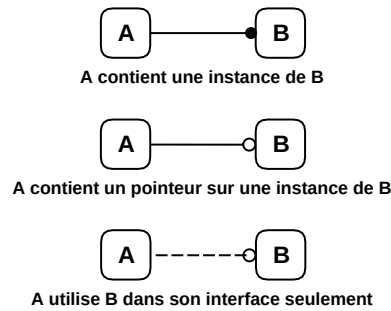


FIG. A.2: Schéma des différents types de relations utilisées dans la figure A.1.

reconstruction géométrique et propose quelques corrections.

Les textures des polygones de la scène reconstruites sont calculées dans la même session mais sont sauvegardées immédiatement dans un fichier sans l'attribuer au polygone correspondant. La manipulation de la scène complète avec les textures implique l'utilisation de beaucoup de mémoire ce qui compromet l'efficacité du système surtout lors des tâches gourmandes en mémoire comme la reconstruction des textures.

## A.2 Système de reconstruction des textures

Le processus de reconstruction des textures dans notre système est relativement indépendant des autres tâches de reconstruction géométrique. De façon générale, il aurait pu constituer une application à part entière complètement détachée du système actuel de reconstruction géométrique. Les interfaces graphiques et les nombreuses librairies du système déjà implémentées nous ont amenés à choisir la fusion plutôt que la scission du processus de reconstruction des textures avec le processus de reconstruction de la géométrie. Ceci nous donne un procédé de reconstruction plus complet qui peut s'effectuer dans une seule session de travail plus uniforme. Le processus de reconstruction des textures englobe deux grandes étapes : l'extraction des données texturales de la scène et l'unification finale de ces données.

Les concepts sous-jacents au processus de reconstruction des textures sont présentés au chapitre 4. Cette section se veut une récapitulation appliquée à l'informatique de ces concepts. Bien que nous allons passer outre plusieurs éléments d'implantation du système, nous porterons une attention particulière aux aspects les plus importants de sorte qu'une révision future des divers mécanismes concernés par notre méthode de

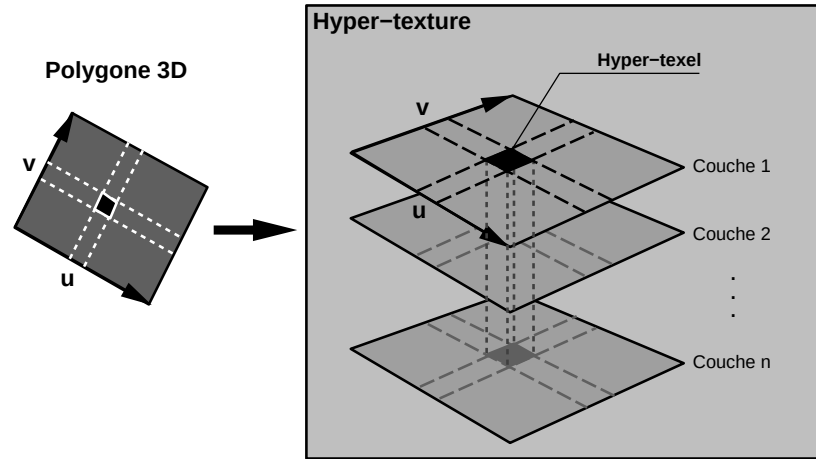


FIG. A.3: Une représentation schématique d’une “hyper-texture”.

reconstruction des textures soit simple et rapide.

Dans un premier temps, nous présenterons en détails la structure de données principale du système de reconstruction des textures. Par la suite, nous exposerons les principaux algorithmes utilisés pour faire l’extraction des données et l’unification des ces dernières.

### A.2.1 Structures de données

Le système de reconstruction des textures implique une structure de données particulière pour stocker et manipuler les différentes informations extraites des images et de la scène (voir la section 4.2). Nous avons nommé cette structure `WsHyperTextureMap` parce qu’elle représente non pas une seule couche de données de couleurs formant une texture mais un ensemble de couches de diverses données. La figure A.3 nous montre la structure sous un aspect schématique.

Plus simplement, on peut considérer l’“hyper-texture” comme étant formée d’une seule couche d’“hyper-texels” qui sont à leur tour formés des informations provenant des diverses images contenant la projection du polygone tridimensionnel. Chaque couche de l’“hyper-texture” sur le schéma représente les informations provenant d’une image.

Chaque texel d’une `WsHyperTextureMap` est un `WsHyperTexel` et comporte toutes les données pertinentes à l’unification. La figure A.4 nous montre la structure interne d’un “hyper-texel”.

La structure de données ainsi constituée nous permet de stocker une variété de

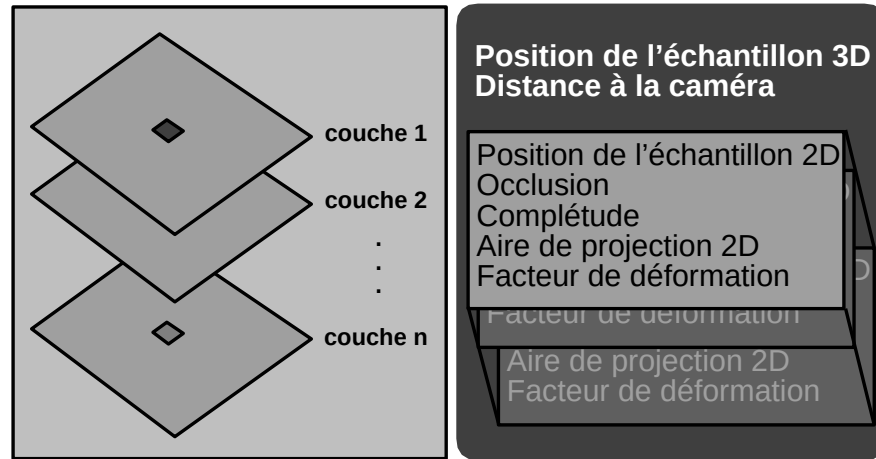


FIG. A.4: Une représentation schématique d'un "hyper-textel".

données pertinentes à l'unification tout en conservant la totalité des informations contenues dans les images et dans la scène. Ofek et al. [OSRW97] composent une structure hiérarchique (pyramidale) à partir des données provenant d'images de différentes qualités et résolutions. Les données étant unifiées directement dans leur niveau de résolution, il y a une corruption graduelle de l'information à mesure que la structure se remplit et que l'information est propagée aux niveaux supérieurs et inférieurs s'il y a lieu. Cette structure peut donc être moins coûteuse en espace mémoire mais apparaît beaucoup moins versatile que la nôtre. De plus, l'étape où on effectue l'échantillonnage et l'extraction des données est la plus coûteuse en temps, et on doit l'effectuer peu importe la structure de stockage utilisée. En conséquence, nous croyons préférable et plus pertinent de conserver les données durement acquises telles qu'elles sont étant donné que notre processus d'unification des texels est moins automatisé et permet l'intégration facile de nouveaux algorithmes et de nouvelles propriétés pour les texels.

### A.2.2 Extraction des données

Une fois que la reconstruction géométrique est terminée, nous sommes en possession d'une scène remplie de polygones sur lesquels il faut apposer des textures. La première étape de ce processus de reconstruction des textures est l'extraction des données des images ayant servi à la reconstruction géométrique. Comme nous l'avons vu à la section 4.2, le processus d'extraction des données est automatique mais est le plus coûteux en temps.

Même s'il serait facile d'extraire toutes les textures pour tous les polygones de la scène directement sans interruption, nous avons décidé de laisser l'utilisateur du système choisir le polygone pour lequel il veut créer la texture, un à la fois. En fait, il serait inutile d'effectuer la longue tâche d'extraction des données pour tous les polygones et de faire attendre l'utilisateur jusqu'au processus d'alignement et d'unification des textures. Ainsi, l'utilisateur choisit un polygone dans une image et le processus d'extraction des données est déclenché.

L'algorithme 1 nous montre l'implantation de notre moteur d'échantillonnage des données dans les images.

---

**Algorithme 1:** Extraction et d'échantillonnage des données

---

**Données :**

$S = \{s_1, s_2, \dots, s_n\}$  : la liste des  $n$  images  $s_i$  du système

$P = \{p_1, p_2, \dots, p_m\}$  : la liste des  $m$  polygones  $p_i$  du système

$index$  : l'index du polygone à échantillonner

$l, h$  : la hauteur et la largeur de l'"hyper-texture" à produire

**Résultat :** "Hyper-texture"  $T$  de résolution  $l \times h$

**début**

Création de  $T$

**pour chaque**  $j \in \{1, 2, \dots, h\}$  **faire**

**pour chaque**  $i \in \{1, 2, \dots, l\}$  **faire**

$pos3D \leftarrow \text{calculerPosition3D}(p_{index}, i, j)$

**pour chaque**  $k \in \{1, 2, \dots, n\}$  **faire**

$T(i, j, k) \leftarrow \text{calculerEchantillon}(s_k, pos3D)$

**fin**

---

Pour chaque texel  $(i, j)$ , on calcule sa position tridimensionnelle ( $pos3D$ ) sur le polygone à l'aide de la fonction  $\text{calculerPosition3D}(\text{polygone}, i, j)$ . Par la suite, pour chaque image, on calcule un échantillon de couleur en projetant  $pos3D$  dans l'image  $s_k$ . Plusieurs exemplaires de la fonction  $\text{calculerEchantillon}(\text{image}, pos3D)$  sont disponibles avec des filtres d'échantillonnage différents. Bien qu'on utilise exclusivement un échantillonnage ponctuel simple dans le cadre de ce projet étant donné qu'on fait un sur-échantillonnage dans les images, il reste qu'on pourrait facilement varier les résultats des

échantillonnages par l'utilisation d'autres filtres. La raison principale pour ne pas faire de filtrage particulier sur les données durant l'extraction de ces dernières des images est qu'on veut éviter toute corruption des données avant d'en faire le traitement. Nous croyons qu'il est préférable de stocker les données brutes dans la structure plutôt que de tenter de simuler le comportement parfois complexe des filtres et de se confondre entre l'information réelle des images et celle obtenue par un filtrage quelconque.

En plus de déterminer la couleur de l'échantillon, la fonction *calculerEchantillon(image, pos3D)* calcule s'il y a une occlusion dans l'échantillon ou si l'échantillon fait partie de l'image. Aussi, on évalue la proportion du pixel échantillonné qui se trouve à l'intérieur et à l'extérieur du polygone d'échantillonnage. Comme on l'a déjà montré, la résolution des images est discrète et les couleurs de la frontière des polygones dans ces dernières sont souvent mélangées avec d'autres couleurs provenant de l'arrière-plan, etc. La proportion ainsi calculée peut nous aider à attribuer une importance plus grande aux pixels de couleurs entièrement contenues dans le polygone d'échantillonnage. Ce mélange des couleurs ne s'effectue pas seulement aux frontières des polygones. Selon l'angle de vue d'un polygone dans une image et sa distance à la caméra, il peut y avoir une bonne partie des couleurs de ce dernier qui se mélangent dans un pixel de l'image. C'est pourquoi nous croyons plus pertinent le calcul de l'aire et du ratio de projection d'un échantillon dans une image pour évaluer la précision ou l'hétérogénéité de sa couleur. Ces données recueillies pour chaque texel sans oublier l'angle de vue et la distance à la caméra pour chaque image, seront utilisées dans l'algorithme d'unification. Les divers concepts sous-jacents au processus d'extraction des données dans les images sont discutés plus précisément à la section 4.2.

### A.2.3 Unification

Le processus d'unification des textures s'effectue en deux étapes : l'alignement des données et la sauvegarde d'une texture unique obtenue par l'algorithme d'unification. Pour l'unification des textures, nous avons uniquement besoin de l'"hyper-texture" créée lors de l'étape d'extraction des données (voir la section A.2.2).

L'étape d'alignement des données consiste à attribuer une matrice de transformation à chaque couche de l'"hyper-texture". La position des texels de la texture finale est

transformée par cette matrice pour que l'unification des “hyper-texels” soit faite sur les “hyper-texels” alignés. Si les diverses couches de l’“hyper-texture” sont suffisamment alignées aux yeux de l'utilisateur, la matrice identité sera attribuée à chaque couche de sorte que tous les “hyper-texels” resteront à leur place durant l'unification. L'algorithme 2 illustre le processus général d'unification des données. Le déroulement de cette étape implique que les matrices d'alignement des couches de l’“hyper-texture” ont été calculées. La résolution de la texture finale est la même que celle de l’“hyper-texture” ayant servi à la créer.

---

**Algorithme 2:** Unification des données

---

**Données :**

$T$  : “hyper-texture” de résolution  $l \times h$  avec  $n$  couches

$M = \{m_1, m_2, \dots, m_n\}$  : la liste des matrices de transformations

$m_i$  pour chaque couche  $k$  de l’“hyper-texture”

**Résultat :** Texture  $I$  de résolution  $l \times h$ .

**début**

Création de  $I$

**pour chaque**  $j \in \{1, 2, \dots, h\}$  **faire**

**pour chaque**  $i \in \{1, 2, \dots, l\}$  **faire**

        Création de la liste des “hyper-texels”  $H = \{h_1, h_2, \dots, h_n\}$

**pour chaque**  $k \in \{1, 2, \dots, n\}$  **faire**

$pos2D \leftarrow transformerPosition(i, j, m_k)$

$h_k \leftarrow T(pos2D)$

$I(i, j) \leftarrow calculerCouleur(H)$

**fin**

---

Pour chaque texel de la texture finale, on forme une liste  $H$  des “hyper-texels” transformés pour faire l'unification et trouver une couleur unique. La position  $pos2D$  de l’“hyper-texel” dans chaque couche  $k$  est obtenue par la fonction  $transformerPosition(i, j, m_k)$ . La position est obtenue de la façon suivante :

$$transformerPosition(x, y, M) = \begin{bmatrix} M_1 & M_2 & M_3 \\ M_4 & M_5 & M_6 \\ M_7 & M_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (A.1)$$

Parfois, l'unification s'effectue sur un nombre plus réduit d’“hyper-texels” (la cardinalité de  $H$  est plus petite que  $n$ ) car il se peut qu'une position transformée  $pos2D$  se

retrouve à l'extérieur de l'“hyper-texture” et par conséquent, ne corresponde à aucun “hyper-textel”.

La fonction *calculerCouleur*( $H$ ) est la fonction d'unification des données. Nous avons analysé une multitude de fonctions utilisant chacune différentes combinaisons des données contenues dans les “hyper-textels”. Les fonctions d'unification des données utilisées ainsi que les résultats qu'elles ont produits sont exposés dans la section 4.3.2.

#### A.2.4 Remplissage

Lors de l'unification des “hyper-textels”, on est parfois confronté à la situation où il nous est impossible de reconstruire une partie de l'information dans une texture. Ceci survient, par exemple, quand une partie d'une surface est cachée dans toutes les images qui contiennent cette surface. Ainsi, les “hyper-textels” à unifier seront tous identifiés comme faisant partie d'une occlusion et aucune couleur ne pourra logiquement être retournée. Puisque les “hyper-textels” dans l'occlusion comportent nécessairement les couleurs d'autres polygones de la scène vus de différents points de vue, il est forcément inutile d'unifier ces couleurs en une seule en espérant que cette dernière ne soit pas trop visible dans la texture finale. Pour les surfaces dont la texture est de couleur uniforme, nous pouvons remplir les trous d'informations manquantes à l'aide de notre algorithme de remplissage. Plusieurs techniques ont été développées pour faire le remplissage des trous, pour créer des textures complexes ou pour éliminer le bruit ou certains éléments indésirables dans les images [HB95][HT96]. Ces techniques procèdent souvent par analyse spectrale de la texture trouée pour combler les parties manquantes. Cependant, ces techniques de remplissage sont plus appropriées pour les textures comportant des motifs bruités et flous, enfin, des textures de distribution plus ou moins aléatoires. Une texture à motifs bien définis et réguliers pourra difficilement être simulée par ces techniques. Les problèmes de trous dans les textures ou plus généralement dans les données sont fréquents pour le rendu à base d'image. En ce sens, les techniques de remplissage pourraient être utilisées pour combler les informations manquantes entre les divers échantillons lors d'une reprojexion (voir la section 2.1). Dans notre cas, nous avons utilisé une méthode très simple qui donne des résultats acceptables pour les textures de couleur uniforme ou légèrement bruitées. Debevec et al. [DBY98] présentent aussi une technique de remplissage des “trous” causés par les informations manquantes

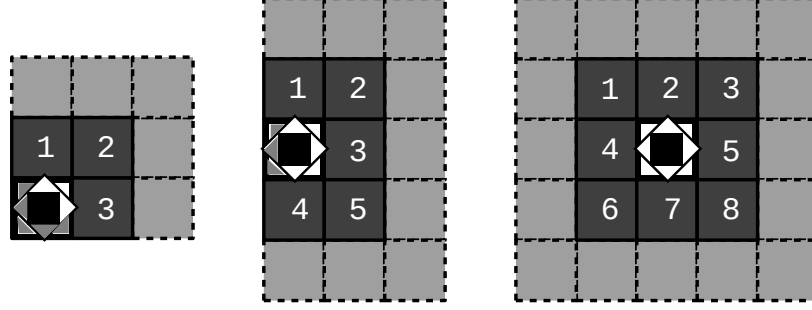


FIG. A.5: Les trois configurations de voisinage d'un texel.

dans les images. Bien que le remplissage soit effectué dynamiquement selon le point de vue dans lequel on fait le rendu de la scène, cette technique comporte des éléments intéressants, comme l'interpolation des couleurs sur les polygones cachés entre autres, qui pourraient améliorer notre technique de remplissage.

Un texel  $(i, j)$  a trois, cinq ou huit voisins, tout dépendant s'il se trouve dans un coin, sur la bordure ou au milieu d'une texture (voir la figure A.5). Pour chaque texel incomplet dans la texture on vérifie ses voisins. Si aucun de ses voisins n'est complet, on passe au texel incomplet suivant. Sinon on fait la moyenne des couleurs de ses voisins, sans compter la couleur des voisins incomplets bien entendu. Le texel devient donc complet et l'algorithme est garanti de terminer. On effectue ces étapes jusqu'à ce qu'il n'y ait plus de texels incomplets.

Le dessus de la table présentée dans la figure A.6 comporte des zones d'occlusions qu'on ne peut reconstruire sous aucun point de vue. Heureusement, sa surface est d'une couleur uniforme et les nombreux "hyper-texels" totalement incomplets, correspondant aux zones d'occlusion causées par les objets directement déposés sur la table, ont pu être interpolés par notre algorithme. Cependant, nous pouvons constater que la photographie posée sur la table comporte elle aussi des zones d'occlusions permanentes qui seront très difficiles à remplir correctement. La méthode utilisée dans ce projet pour déterminer les occlusions dans une texture et les problèmes qui y sont reliés sont discutés à la section 4.2.2.

Le problème de remplissage est un problème excessivement complexe car il consiste à extrapoler des informations qui ne sont pas dans les images. La méthode que nous utilisons, tout comme la plupart de autres méthodes de remplissage, interpole un résultat en faisant l'hypothèse qu'une couleur manquante est effectivement rapprochée de l'in-

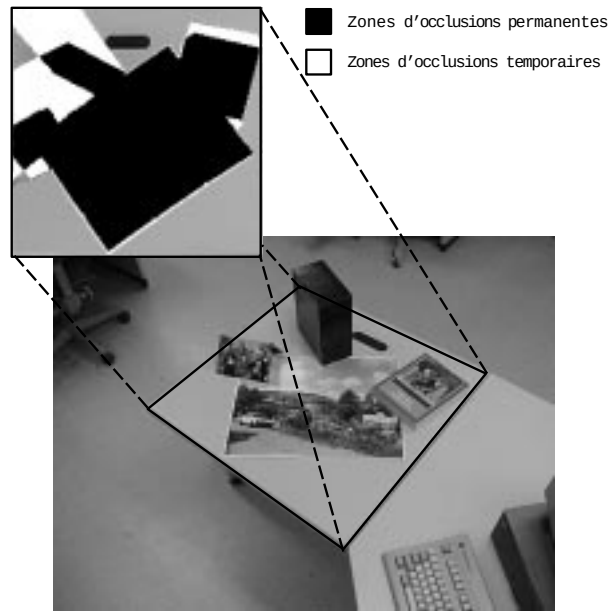


FIG. A.6: Les zones d'occlusion temporaire et permanente sur une table.

terpolation de ses voisins. Pour les textures de hautes fréquences et qui comportent beaucoup d'informations sans motifs répétitifs, un remplissage satisfaisant ne peut s'effectuer sans l'intervention humaine.

### A.3 Système général

L'implantation d'un système de reconstruction complet peut sembler directe et simple d'un certain point de vue, mais elle comporte plusieurs petits détails qui complexifient le développement. Un tel système implique l'obtention d'un résultat complet, d'une scène en l'occurrence, décrite dans une représentation permettant un rendu infographique. La scène obtenue devra comporter toutes les informations pertinentes pouvant être extraites des images. Dans notre cas, bien que nos scènes reconstruites comportent certaines lacunes au niveau des propriétés des surfaces et des sources de lumière, nous considérons que la reconstruction géométrique et texturale est suffisamment complète pour produire des modèles pouvant être rendus avec une grande variété de techniques en infographie.

Comme dans plusieurs projets de recherche requérant une implantation logicielle, plusieurs décisions sur la portabilité, la clarté, l'efficacité et la flexibilité de l'implantation sont laissées pour compte en dépit de leur grande influence sur la solidité des

arguments de recherche impliqués. Bien qu'on puisse facilement écarter ces décisions du processus de recherche en les qualifiant d'esthétiques, il reste que souvent, elles sont cruciales pour l'avancement cohérent d'un projet. De plus, ces décisions d'implantation donnent souvent un appui non négligeable à la théorie même si elles sont habituellement éloignées conceptuellement.

On pourrait facilement écrire plusieurs livres entièrement fondés sur la recherche scientifique et les problèmes d'implantation informatique qui y sont reliés. Cette section se veut une présentation plus critique du système en mettant en lumière ses erreurs et les corrections possibles à ces dernières. Cet exposé des difficultés de l'implantation pourra sembler inutile et superflu pour certains, mais nous espérons qu'il pourra éviter certains pièges à d'autres qui seront peut-être confrontés aux mêmes questionnements lors de l'implantation d'un tel système. On pourra évoquer cette section comme une discussion finale sur la bonne marche du projet de ce travail.

### A.3.1 Analyse et design

Plusieurs détails au sujet des structures de données doivent être pris en considération lors de la conception d'un système de reconstruction. Contrairement à la philosophie reflétée dans le système actuel, nous croyons qu'un système de reconstruction polyvalent et extensible doit être fondé sur un ensemble minimal de structures de données dont les composantes et les comportements sont simples et restreints. Le système de reconstruction résultant de ce travail est loin d'évoquer la simplicité. Il reste toutefois que cette complexité est engendrée par la diversité des concepts qui sont intégrés dans le système. La difficulté de formalisation et de traduction de ces concepts en système informatisé peut justifier en grande partie la lourdeur de certaines parties de notre logiciel.

À notre avis, il est possible de créer une architecture d'un système de reconstruction qui soit à la fois concise et versatile. Dans le cas du système de reconstruction géométrique, il faut véritablement réduire l'interaction de toutes sortes de données avec le moteur de convergence. Nous croyons que la structure géométrique fondamentale de la reconstruction est le point bidimensionnel se trouvant dans les images et que la scène reconstruite n'est que le résultat d'un ensemble de relations entre ces points. Ainsi, tout ce que l'utilisateur entre dans le système via l'interface graphique est traduit

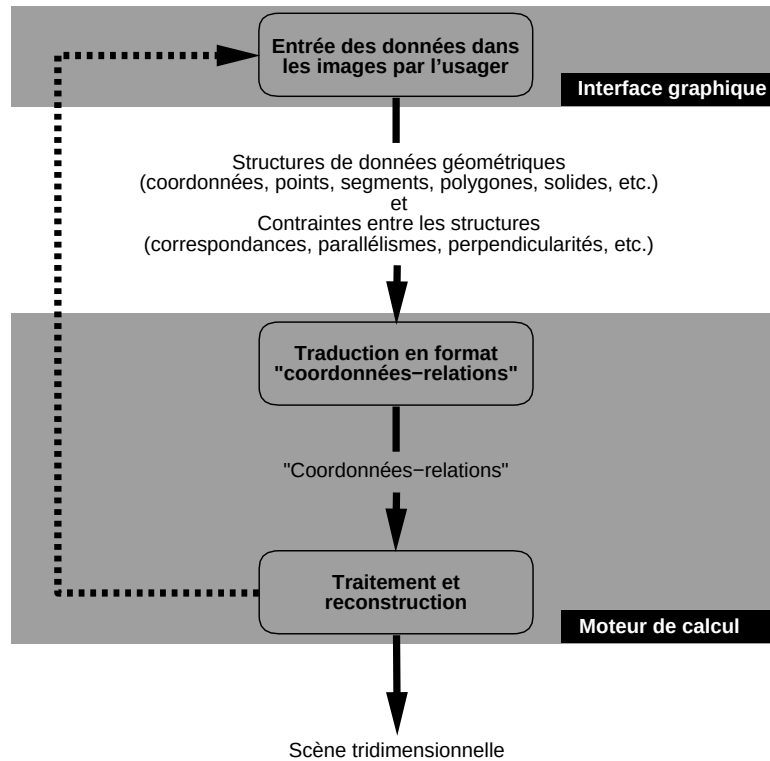


FIG. A.7: Schéma du processus de reconstruction géométrique.

en termes de points  $2D$  et  $3D$  et de relations entre ces derniers. On remarque toutefois que, bien qu'un point bidimensionnel est l'unité la plus élémentaire pouvant être utilement manipulée par l'utilisateur, elle impose une certaine limite sur la variété des contraintes utiles pouvant être exprimées dans un système de reconstruction. En fait, ce qui importe dans le système de reconstruction c'est la versalité et la polyvalence du système à pouvoir intégrer une multitude de relations différentes sur les données. On remarque en fait, que la puissance d'expression requise pour spécifier les contraintes indispensables au système de reconstruction s'applique à la base aux coordonnées individuelles et non aux points qui sont un ensemble de deux ou trois coordonnées. Un groupe de contraintes utiles pour exprimer l'adjacence ou la distance de deux primitives par exemple, ne peuvent être exprimées facilement sans traiter les coordonnées individuellement.

Le nouveau processus de reconstruction géométrique basé sur les coordonnées est illustré à la figure A.7. D'énormes avantages sont amenés par cette nouvelle conception. Entre autres, il serait très facile d'y exprimer de nouvelles contraintes sur les données. En fait, toutes fonctions mathématiques sur les coordonnées réelles ( $\in \mathbb{R}$ ) pourraient

servir de contraintes. Le pouvoir d'expression des contraintes dans le système serait donc illimité. L'intégration de nouvelles contraintes dans le système actuel est une tâche complexe car celles-ci peuvent s'appliquer sur un ensemble de structures hétérogènes et car chacune fait une action particulière qui ne peut être généralisée. Ceci provoque un décuplement des classes de contraintes proportionnellement au nombre de classes de données et aux relations entre ces dernières. Dans le système basé sur les coordonnées, les contraintes sont des fonctions mathématiques que l'utilisateur peut exprimer dynamiquement par l'entremise d'un éditeur et d'un interprète spécialisé. De plus, la définition des modèles est beaucoup plus simple parce qu'on n'a qu'à spécifier une fonction d'échantillonnage des modèles en coordonnées ou en points  $2D$  pour les intégrer dans le système. Ceci implique en fait qu'on puisse représenter les modèles par des échantillons de leur surface.

Snyder et Kajiya [SK92] décrivent une technique intéressante pour décrire des modèles de façon symbolique. Les modèles ainsi décrits se nomment modèles par balayage ou cylindres généralisés. Une surface par balayage est obtenue en déplaçant (balayant) une tranche courbe (*cross section*) le long d'une trajectoire courbe (*path*). Certaines méthodes de reconstruction géométrique, dont celle présentée par Flesia et Roche [FR92], utilisent les modèles par balayage parce qu'ils sont plus faciles à détecter et à reconstruire automatiquement dans les images. Dans les systèmes plus interactifs où on privilégie l'intervention de l'utilisateur, ce dernier n'a qu'à entrer la tranche génératrice et la trajectoire courbe dans les images. Ces modèles définis par deux courbes sont faciles à manipuler et leur surface peut être facilement échantillonnée ou polygonisée avec différentes résolutions. Cette technique de description des modèles pourrait très bien s'appliquer à l'architecture basée sur les coordonnées et pourrait faciliter son automatisation. La technique de Snyder et Kajiya permet de créer des modèles en spécifiant des équations paramétriques pour la trajectoire de balayage et la tranche génératrice. La spécification des équations des modèles et des équations pour les contraintes pourrait simplement s'uniformiser pour donner un langage de reconstruction géométrique complet et puissant.

Pour ce qui est de la reconstruction des textures, nous croyons que cette étape constitue le parachèvement simple et direct de l'effort qu'il a fallu pour déduire la structure géométrique des images. L'extraction des données est automatique et leur

traitement pourrait s'allier plusieurs automatismes qui nous permettraient de retrancher la nécessité des interventions humaines. La conclusion de ce travail (voir le chapitre 5) discute de ces améliorations à propos du processus de reconstruction texturale. Sans aucun doute, les primitives à échantillonnage adaptatif de Snyder et Kajiya en conjonction avec l'uniformité du traitement des relations et des contraintes entre ces dernières, pourrait certainement simplifier et amplifier l'apport des textures au réalisme de la reconstruction. Plus simplement, on pourrait faire correspondre les couleurs des points sur une surface et spécifier des contraintes entre ces dernières basées sur l'illumination et les diverses propriétés de surface par exemple. Finalement, les coordonnées, les couleurs et toutes les autres propriétés d'un point seraient toutes intégrées pour pouvoir interagir entre elles de façon transparente et uniforme. Les structures de données utilisées dans ce système pour la reconstruction des textures, reflètent une technique embryonnaire qui a pour seul but l'exploration de certaines avenues pour améliorer les textures provenant d'images. Cette technique vorace, est vouée à de nombreuses mutations comme énoncées dans la conclusion de ce travail. Soit qu'elle devienne une étape finale et servile de sélection esthétique faite par un usager. Soit qu'on explore plus avant les possibilités qui s'offrent au niveau de la reconstruction de l'illumination et des propriétés de surface.

### A.3.2 Programmation

Une fois que l'architecture du système est décrite formellement, ses qualités et ses failles transparaissent directement dans son implantation. Les structures de données de notre système sont organisées de façon à complexifier leurs comportements et ceci se transcrit visiblement dans le code source du système. La lourdeur des structures provient majoritairement du fait qu'on effectue une gestion principalement manuelle de ces dernières. Ceci affecte plus ou moins le caractère direct des calculs du moteur de convergence sur les contraintes. Le processus général de calcul des contraintes par résolution d'un système d'équations linéaires est simple et bien cerné dans l'implantation. Les résultats qui proviennent du système sont faciles à tester parce que les fonctions de calculs sont bien centralisées. Ce sont les interactions variées de l'utilisateur avec le système et la conception vague et abstraite de la notion de contrainte qui confèrent une grande complexité aux structures de l'implantation. C'est certainement le tribut d'une implan-

tation dédiée à la recherche scientifique; elle fut pressée par le temps et limitée par la réflexion et les questionnements.

L'expérience acquise lors du développement de notre système de reconstruction nous inspire confiance quant à la pertinence d'une restructuration plus simple et plus succincte d'un système de reconstruction complet. Cette nouvelle architecture comporte sûrement une multitude de problèmes imprévus, mais elle contourne avantageusement ceux qui proviennent de l'implantation antérieure.

## Annexe B

# Méthode des moindres carrés

Il est possible de transformer les texels d'une texture  $A$  dans les texels d'une texture  $B$  en calculant une matrice de transformation entre les deux textures et en la multipliant avec la position des texels de la texture  $A$ . Si nous avons au moins trois paires de texels correspondants sur chaque texture, il est possible de trouver une matrice de transformation avec la méthode des moindres carrés.

Soit  $q$  un texel d'une texture  $A$  et  $p$  un texel d'une texture  $B$ . Nous voulons la matrice  $M$  telle que

$$q = M \cdot p^T. \quad (\text{B.1})$$

Si nous avons  $n$  texels correspondants dans chaque texture alors on peut construire les matrices  $Q$  et  $P$  telles que

$$Q = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} = \begin{bmatrix} q_{1x} & q_{2x} & \cdots & q_{nx} \\ q_{1y} & q_{2y} & \cdots & q_{ny} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \quad (\text{B.2})$$

$$P = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix} = \begin{bmatrix} p_{1x} & p_{2x} & \cdots & p_{nx} \\ p_{1y} & p_{2y} & \cdots & p_{ny} \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \quad (\text{B.3})$$

Évidemment, cette méthode implique l'existence d'au moins trois corrélations entre les points. Donc si  $n \geq 3$  on a

$$e_i = q_i - M \cdot p_i \quad (\text{B.4})$$

$$E = \begin{bmatrix} e_1 & e_2 & \cdots & e_n \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & \cdots & e_{nx} \\ e_{1y} & e_{2y} & \cdots & e_{ny} \\ 1 & 1 & \cdots & 1 \end{bmatrix}. \quad (\text{B.5})$$

Après certaines dérivations (voir la démonstration fournie par Horn [Hor86]) on obtient

$$M = Q \cdot P \cdot (P \cdot P^T)^{-1}. \quad (\text{B.6})$$

## Annexe C

# Calcul de la caméra d'une image

Ce chapitre présente notre méthode pour calculer la position et la direction de vue d'une caméra pour une image. Ces paramètres nous servent à déterminer la visibilité des primitives projetées dans une image grâce à la technique du lancer de rayons. De plus, ils nous permettent d'évaluer la distance d'un point d'échantillonnage à la caméra et l'angle entre la direction de vue et la surface de sorte qu'on puisse attribuer un plus grand poids aux échantillons qui sont susceptibles d'être plus précis (voir la section 4.2.2). En fait, la position tridimensionnelle et la direction de vue de la caméra pour une image sont indispensables pour calculer une bonne partie des attributs liés à la situation spatiale d'un échantillon tridimensionnel sur une primitive par rapport à une image.

### C.1 Calcul de la position d'une caméra

Une matrice de projection obtenue par la reconstruction d'au moins six points dans une image (voir la section 3.4), est de la forme suivante :

$$T = \begin{bmatrix} T_0 & T_1 & T_2 & T_3 \\ T_4 & T_5 & T_6 & T_7 \\ 0 & 0 & 0 & 0 \\ T_8 & T_9 & T_{10} & 1 \end{bmatrix}. \quad (\text{C.1})$$

Posons une ligne bidimensionnelle  $l$  passant par  $p_1$  et  $p_2$ , deux points dans une image, (voir la figure C.1) et ayant la forme suivante :

$$l = \begin{bmatrix} a & b & c \end{bmatrix}^T = \begin{bmatrix} p_{1y} - p_{2y} & p_{1x} - p_{2x} & p_{1x}p_{2y} - p_{1y}p_{2x} \end{bmatrix}^T. \quad (\text{C.2})$$

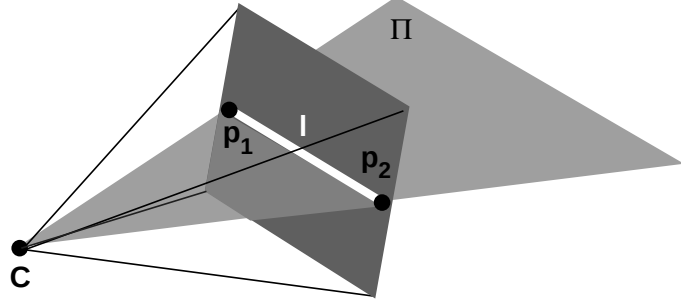


FIG. C.1: La ligne  $l$ , passant par les points  $p_1$  et  $p_2$  dans l'image, forme un plan avec  $C$ , le centre de projection de la caméra de l'image.

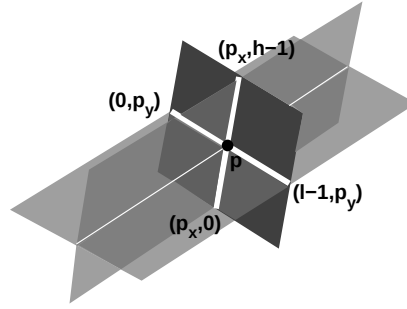


FIG. C.2: Deux plans formés à un point dans une image par les lignes verticale et horizontale passant par ce point.

Alors, d'après Poulin et al. [POF98], l'équation du plan  $\Pi$  passant par  $C$  et par  $l$  est obtenue comme suit :

$$\Pi = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = l \cdot T = \begin{bmatrix} a & b & \star & c \end{bmatrix} \begin{bmatrix} T_0 & T_1 & T_2 & T_3 \\ T_4 & T_5 & T_6 & T_7 \\ 0 & 0 & 0 & 0 \\ T_8 & T_9 & T_{10} & 1 \end{bmatrix} = \begin{bmatrix} aT_0 + bT_4 + cT_8 \\ aT_1 + bT_5 + cT_9 \\ aT_2 + bT_6 + cT_{10} \\ aT_3 + bT_7 + c \end{bmatrix}. \quad (C.3)$$

Il y a deux plans pour chaque point dans une image : le plan défini par la ligne horizontale passant par le point et le plan défini par la ligne verticale passant par le point (voir la figure C.2). L'intersection des plans d'au moins deux points ne donne pas un point tridimensionnel exact ; on utilise la méthode des moindres carrés (voir l'annexe B) pour trouver le point d'intersection "moyen" entre tous les plans. Si on choisit certains points dans une image, et qu'on fait l'intersection des plans définis à chacun de ces points ensemble, on obtient la position de la caméra de l'image à l'endroit où tous les plans convergent.

Une fois que la position de la caméra est calculée nous l'utilisons pour établir la

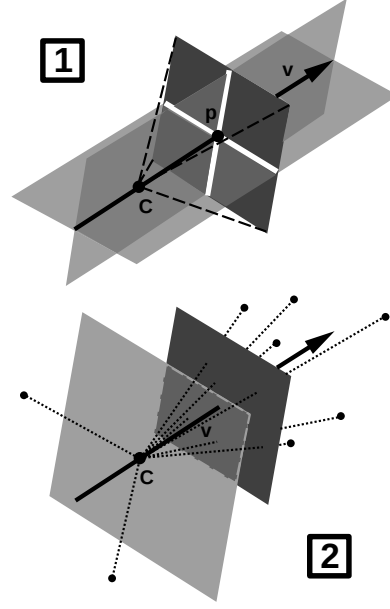


FIG. C.3: Les étapes du calcul de la direction de vue d'une caméra.

direction de vue. Les étapes de calcul de la direction de vue sont montrées à la figure C.3.

Dans un premier temps (étape 1), on prend le point central de l'image<sup>1</sup> et on trouve les plans vertical et horizontal passant par ce point (voir la section C.1). L'intersection des deux plans nous donne la droite de support de la direction de vue. Le vecteur direction  $\vec{v}$  de la droite peut soit être dans le sens de vue ou en sens inverse. Pour déterminer le vrai sens de la direction de vue (étape 2), on trouve le plan parallèle au plan image passant par la caméra et dont la normale est  $\vec{v}$ . Pour tous les points tridimensionnels reconstruits avec l'image, c'est-à-dire les points reconstruits qui projettent dans les limites de l'image, on détermine s'ils sont devant ou derrière le plan de la caméra. Le côté du plan comportant le plus de points détermine le sens du vecteur.

Évidemment, il peut sembler inutile de tester tous les points pour trouver le sens du vecteur direction car ces points doivent logiquement tous se trouver à l'avant de l'image. En fait, bien qu'il soit rare qu'un point reconstruit dans une image se retrouve derrière le plan de cette dernière, certaines erreurs de précision peuvent causer l'inversion de la position des points par rapport au plan image de sorte qu'il faut tester tous les points pour éliminer l'intervention des mauvais points. Ces derniers cas pourraient

<sup>1</sup> $p = (l/2, h/2)$  où  $l$  est la largeur de l'image et  $h$  est la hauteur de l'image.

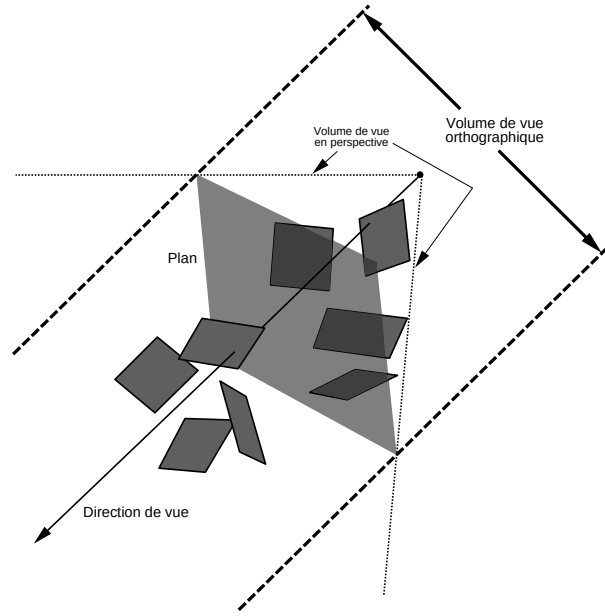


FIG. C.4: Un plan pour remplacer le point de vue orthographique.

survenir si on a une vue orthographique dont le centre de projection tend vers l'infini par exemple. Bien entendu, notre méthode est erronée pour les cas où on aurait plus de mauvais points que de bons, mais ces cas dénoteraient un mauvais fonctionnement général du système et il nous semble extrêmement improbable que ce genre de situations surviennent. Pour ce qui est du problème de la caméra orthographique (voir la figure C.4) qui nous donnerait une point de vue à l'infini, on pourrait déplacer un plan le long de la droite de support de la direction de vue et établir la position de ce dernier pour qu'il soit entièrement derrière tous les objets présents dans l'image. Ce plan nous donnerait une description de caméra plus complète et moins sujette aux erreurs que celle composée de la position et de la direction de vue.

# Bibliographie

- [Bli78] James F. Blinn. « Simulation of Wrinkled Surfaces ». In *Computer Graphics (SIGGRAPH '78 Proceedings)*, volume 12, pages 286–292, août 1978.
- [BN76] James F. Blinn et Martin E. Newell. « Texture and Reflection in Computer Generated Images ». *Communications of the ACM*, volume 19, pages 542–546, 1976.
- [BN92] Thaddeus Beier et Shawn Neely. « Feature-based image metamorphosis ». In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 35–42, juillet 1992.
- [CG85] Rikk J. Carey et Donald P. Greenberg. « Textures for Realistic Image Synthesis ». *Computers and Graphics*, volume 9, numéro 2, pages 125–138, 1985.
- [Che95] Shenchang Eric Chen. « Quicktime VR - An Image-Based Approach to Virtual Environment Navigation ». In *Computer Graphics (SIGGRAPH '95 Proceedings)*, volume 29, pages 29–38, août 1995.
- [CLF98] Emilio Camahort, Apostolos Leros et Donald Fussell. « Uniformly Sampled Light Fields ». In *Proceedings of Eurographics Workshop on Rendering 98*, pages 117–130, juin 1998.
- [Coo84] Robert L. Cook. « Shade trees ». In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 223–231, juillet 1984.
- [Cro84] Franklin C. Crow. « Summed-area Tables for Texture Mapping ». In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 207–212, juillet 1984.
- [DBY98] Paul E. Debevec, George Borshukov et Yizhou Yu. « Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping ». In

- Proceedings of Eurographics Workshop on Rendering 98*, pages 105–116, juin 1998.
- [Deb98] Paul E. Debevec. « Rendering Synthetic Objects Into Real Scenes : Bridging Traditional and Image-Based Graphics With Global Illumination and High Dynamic Range Photography ». In *Computer Graphics (SIGGRAPH '98 Proceedings)*, volume 32, pages 189–198, juillet 1998.
- [DM97] Paul E. Debevec et Jitendra Malik. « Recovering high dynamic range radiance maps from photographs ». In *Computer Graphics (SIGGRAPH '97 Proceedings)*, volume 31, pages 369–378, août 1997.
- [DRB97] George Drettakis, Luc Robert et Sylvain Bougnoux. « Interactive Common Illumination for Computer Augmented Reality ». In *Eurographics Rendering Workshop 1997*, pages 45–56, New York City, NY, juin 1997. Eurographics, Springer Wein.
- [DTM96] Paul E. Debevec, Camillo J. Taylor et Jitendra Malik. « Modeling and Rendering Architecture from Photographs : A Hybrid Geometry- and Image-Based Approach ». In *Computer Graphics (SIGGRAPH '96 Proceedings)*, volume 30, pages 11–20, août 1996.
- [FGR93] Alain Fournier, Atjeng S. Gunawan et Chris Romanzin. « Common illumination between real and computer generated scenes ». In *Proceedings of Graphics Interface '93*, pages 254–262, Toronto, Ontario, Canada, mai 1993. Canadian Information Processing Society.
- [FR92] S. Flesia et M. Roche. « 3-D reconstruction of generalized cylinders from biplane projections ». *Computers and Graphics*, volume 16, numéro 2, pages 167–174, 1992.
- [Fra98] Marie-Claude Frasson. « Reconstruction Interactive de Scènes Tridimensionnelles à Partir d'Images ». Mémoire de maîtrise, Département d'Informatique et de Recherche Opérationnelle de l'Université de Montréal, septembre 1998.
- [FS80] B. Fishman et B. J. Schachter. « Computer Display of Height Fields ». *Computers and Graphics*, pages 53–60, 1980.

- [Gar85] Geoffrey Y. Gardner. « Visual Simulation of Clouds ». In *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 297–303, juillet 1985.
- [GGSC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski et Michael F. Cohen. « The Lumigraph ». In *Computer Graphics (SIGGRAPH '96 Proceedings)*, volume 30, pages 43–54, août 1996.
- [GW92] Rafael C. Gonzalez et Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company Inc., Reading, MA, 1992.
- [HB95] David J. Heeger et James R. Bergen. « Pyramid-Based Texture Analysis/Synthesis ». In *Computer Graphics (SIGGRAPH '95 Proceedings)*, volume 29, pages 229–238, août 1995.
- [Hec86] Paul S. Heckbert. « Survey of Texture Mapping ». *IEEE Computer Graphics and Applications*, volume 6, numéro 11, pages 56–67, novembre 1986.
- [Hec89] Paul S. Heckbert. « Fundamentals of Texture Mapping and Image Warping ». Mémoire de maîtrise, Department of Electrical Engineering and Computer Science, University of California, Berkeley, juin 1989.
- [HFD90] Jian Ho, Brian V. Funt et Mark S. Drew. « Separating a Color Signal into Illumination and Surface Reflectance Components : Theory and Application ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 12, numéro 10, pages 7–32, octobre 1990.
- [HL90] Pat Hanrahan et Jim Lawson. « A Language for Shading and Lighting Calculations ». In *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 289–298, août 1990.
- [HLM96] Parag Havaldar, Mi-Suen Lee et Gérard Medioni. « View synthesis from unregistered 2-D images ». In *Proceedings of Graphics Interface '96*, pages 61–69, mai 1996.
- [Hor86] Berthold K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.
- [HT96] Anil N. Hirani et Takashi Totsuka. « Combining Frequency and Spatial Domain Information for Fast Interactive Image Noise Removal ». In *Computer Graphics (SIGGRAPH '96 Proceedings)*, volume 30, pages 269–276, août 1996.

- [KSK88] Gudrun J. Klinker, Steven A. Shafer et T. Kanade. « The Measurement of Highlights in Color Images ». *International Journal of Computer Vision*, volume 2, pages 7–32, 1988.
- [Lak96] John Lakos. *Large-Scale C++ Software Design*. Addison-Wesley, Reading, MA, 1996.
- [LdlFK<sup>+</sup>96] F. Leymarie, A. de la Fortelle, J.J. Koenderink, A.M.L. Kappers, M. Stavrudi, B. van Ginneken, S. Muller, S. Krake, O. Faugeras, L. Robert, C. Gauclin, S. Laveau et C. Zeller. « REALISE : reconstruction of REALity from Image SEquences ». In *International Conference on Image Processing*, pages 651–654, Lausanne (Switzerland), sep 1996. IEEE Signal Processing Society.
- [LFTG97] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance et Donald P. Greenberg. « Non-Linear Approximation of Reflectance Functions ». In *Computer Graphics (SIGGRAPH '97 Proceedings)*, volume 31, pages 117–126, août 1997.
- [LH96] Marc Levoy et Pat Hanrahan. « Light Field Rendering ». In *Computer Graphics (SIGGRAPH '96 Proceedings)*, volume 30, pages 31–42, août 1996.
- [Lip80] Andrew Lippman. « Movie-Maps : an Application of the Optical Videodisc to Computer Graphics ». *Computer Graphics (SIGGRAPH '80 Proceedings)*, volume 14, numéro 3, pages 32–42, juillet 1980.
- [Man88] M. Mantyla. *An Introduction to Solid Modeling*. Computer Science Press, Rockville, Md, 1988.
- [MB95] Leonard McMillan et Gary Bishop. « Plenoptic Modeling : An Image-Based Rendering System ». In Robert Cook, éditeur. *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 39–46. ACM SIGGRAPH, Addison Wesley, août 1995. Held in Los Angeles, California, 06-11 August 1995.
- [MKM89] Kenton F. Musgrave, Craig E. Kolb et Robert S. Mace. « The Synthesis and Rendering of Eroded Fractal Terrains ». In *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 41–50, juillet 1989.

- [NB95] W. Niem et H. Broszio. « Mapping Texture from Multiple Camera Views onto 3D-object Models for Computer Animation ». In *Proceedings of the International Workshop on Stereoscopic and Three Dimensional Imaging*, septembre 1995.
- [OSRW97] Eyal Ofek, Erez Shilat, Ari Rappoport et Michael Werman. « Multiresolution Textures from Image Sequences ». *IEEE Computer Graphics & Applications*, volume 17, numéro 2, pages 18–29, mars-avril 1997.
- [Pea85] Darwyn R. Peachey. « Solid Texturing of Complex Surfaces ». In *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 279–286, juillet 1985.
- [Per85] Ken Perlin. « An Image Synthesizer ». In *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 287–296, juillet 1985.
- [PH89] Ken Perlin et Eric M. Hoffert. « Hypertexture ». In *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 253–262, juillet 1989.
- [Pho] « PhotoModeler ». URL : <http://www.photomodeler.com>.
- [Pho75] Bui-T. Phong. « Illumination for Computer Generated Pictures ». *Communications of the ACM*, volume 18, numéro 6, pages 311–317, juin 1975.
- [POF98] Pierre Poulin, Mathieu Ouimet et Marie-Claude Frasson. « Interactively Modeling with Photogrammetry ». In *Proceedings of Eurographics Workshop on Rendering 98*, pages 93–104, juin 1998.
- [Res96] IBM Research. « Texture Synthesis Project ». URL : <http://www.trl.ibm.co.jp/projects/s7340/texture/textureE.htm>, 1996.
- [SD96] Steven M. Seitz et Charles R. Dyer. « View Morphing : Synthesizing 3D Metamorphoses Using Image Transforms ». In *Computer Graphics (SIGGRAPH '96 Proceedings)*, volume 30, pages 21–30, août 1996.
- [SK92] John M. Snyder et James T. Kajiya. « Generative modeling : A symbolic system for geometric modeling ». In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 369–378, juillet 1992.
- [SKv<sup>+</sup>92] Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran et Paul E. Haeberli. « Fast shadows and lighting effects using texture mapping ».

- In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 249–252, juillet 1992.
- [SN86] A. Scheuing et H. Niemann. « Computing Depth From Stereo Images by Using Optical Flow ». *Pattern Recognition Letters*, volume 4, pages 205–212, 1986.
- [SS97] Richard Szeliski et Heung-Yeung Shum. « Creating Full View Panoramic Mosaics and Environment Maps ». In *Computer Graphics (SIGGRAPH '97 Proceedings)*, volume 31, pages 251–258, août 1997.
- [SWI97] Yoichi Sato, Mark D. Wheeler et Katsushi Ikeuchi. « Object Shape and Reflectance Modeling from Observation ». In *Computer Graphics (SIGGRAPH '97 Proceedings)*, volume 31, pages 379–388, août 1997.
- [WG96] S. Weik et O. Grau. « Recovering 3-D Object Geometry using a Generic Constraint Description ». In *ISPRS96 - 18th Congress of the International Society for Photogrammetry and Remote Sensing*, juillet 1996.
- [Wil83] Lance Williams. « Pyramidal Parametrics ». In *Computer Graphics (SIGGRAPH '83 Proceedings)*, volume 17, pages 1–11, juillet 1983.
- [Wol90] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1990.

