



**HAL**  
open science

# Modélisation et rendu d'images réalistes de paysages naturels

Christophe Chaudy

► **To cite this version:**

Christophe Chaudy. Modélisation et rendu d'images réalistes de paysages naturels. Interface homme-machine [cs.HC]. Université Joseph-Fourier - Grenoble I, 1997. Français. NNT : . tel-00004925

**HAL Id: tel-00004925**

**<https://theses.hal.science/tel-00004925>**

Submitted on 20 Feb 2004

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

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

# THÈSE

PRÉSENTÉE PAR

Christophe CHAUDY

POUR OBTENIR LE TITRE DE DOCTEUR  
DE L'UNIVERSITÉ JOSEPH FOURIER - GRENOBLE 1  
(ARRÊTÉS MINISTÉRIELS DU 5 JUILLET 1984 ET 30 MARS 1992)

SPÉCIALITÉ INFORMATIQUE

## Modélisation et rendu d'images réalistes de paysages naturels

Soutenue le 5 février 1997 devant le jury composé de :

MM. Claude Puech	Président
Philippe Genoud	Directeurs
Jacques Lemordant	
Didier Arquès	Rapporteurs
Bernard Péroche	
Pierre Dumolard	Examineur

THÈSE PRÉPARÉE AU SEIN DU LABORATOIRE GRAVIR-IMAG



## Remerciements

Je remercie Messieurs Didier Arquès et Bernard Péroche d'avoir accepté d'être rapporteurs de cette thèse et Messieurs Claude Puech et Pierre Dumolard pour m'avoir fait l'honneur de participer au jury.

Je remercie également Philippe Genoud et Jacques Lemordant pour m'avoir accueilli et ainsi que toute l'équipe ViS pour son soutien et sa bonne humeur.

Je tiens aussi a remercier Christine, Fred, Mähnu et Philippe qui ont relu mon rapport de thèse avec beaucoup d'attention et d'intérêt.



# Table des matières

<b>I</b>	<b>Introduction</b>	<b>5</b>
	La motivation . . . . .	7
	Les objectifs . . . . .	8
	L'organisation du rapport . . . . .	10
<b>II</b>	<b>État de l'art</b>	<b>13</b>
<b>1</b>	<b>Les objets et les phénomènes naturels</b>	<b>17</b>
1.1	Les végétaux . . . . .	17
1.1.1	Les modèles structurels . . . . .	19
1.1.1.1	Qu'est ce que la structure ramifiée d'un végétal? . . . . .	19
1.1.1.2	Les premiers modèles à caractère géométrique [AK84] . . . . .	21
1.1.1.3	Les modèles fractals [Opp86, Smi84] . . . . .	22
1.1.1.4	Les modèles à base de paramètres botaniques [dREF <sup>+</sup> 88] . . . . .	23
1.1.1.5	Les modèles à base de grammaires [PLH88] . . . . .	26
1.1.1.6	Les modèles combinatoires [VEJA89] . . . . .	31
1.1.1.7	Les modèles récents à caractère géométrique [WP95] . . . . .	33
1.1.2	Les modèles « impressionnistes » . . . . .	36
1.1.2.1	Qu'est ce qu'un modèle impressionniste? . . . . .	36
1.1.2.2	Les modèles à base de textures [Gar84] . . . . .	36
1.1.2.3	Les modèles dans un espace de voxels [Gre89] . . . . .	39
1.1.2.4	Les modèles à base de système de particules [RB85] . . . . .	41
1.1.3	Classification et comparaison . . . . .	43
1.1.4	Conclusion . . . . .	44
1.2	Le relief . . . . .	47
1.2.1	Revue des méthodes pour la modélisation de terrains . . . . .	47
1.2.2	Le rendu des reliefs . . . . .	52
1.2.2.1	Les cartes d'élévation . . . . .	53
1.2.2.2	Les réseaux de polygones . . . . .	54
1.2.3	Conclusion . . . . .	56
1.3	Les phénomènes atmosphériques . . . . .	56

1.3.1	Les nuages et brumes . . . . .	57
1.3.1.1	Les modèles physiques . . . . .	57
1.3.1.2	Les modèles empiriques . . . . .	59
1.3.2	Le ciel . . . . .	60
1.3.3	Pluie et neige . . . . .	62
1.3.4	Conclusion . . . . .	62
<b>2</b>	<b>Techniques de rendu</b>	<b>63</b>
	Visibilité des surfaces . . . . .	63
	Illumination des surfaces et réflexion de la lumière . . . . .	64
2.1	Le Z-buffer et ses extensions . . . . .	65
2.1.1	Ombres et réflexions . . . . .	66
2.1.2	Optimisation du rendu par Z-buffer . . . . .	67
2.2	Conclusion sur les techniques de rendu . . . . .	69
<b>3</b>	<b>Conclusions de l'état de l'art</b>	<b>71</b>
<b>III</b>	<b>Les systèmes de particules pour la modélisation et le rendu</b>	<b>73</b>
<b>1</b>	<b>Introduction</b>	<b>75</b>
<b>2</b>	<b>Définitions</b>	<b>77</b>
<b>3</b>	<b>Utilisation</b>	<b>81</b>
<b>4</b>	<b>Mise en œuvre</b>	<b>85</b>
4.1	Implémentation du système de particules . . . . .	85
4.1.1	La structure PARTICULE . . . . .	86
4.1.2	Les structures ÉTAT et TRANSITION . . . . .	86
4.1.3	Algorithme principal . . . . .	87
4.2	Rendu de particules . . . . .	87
4.2.1	Élimination des parties cachées . . . . .	88
4.2.2	Calcul de la couleur . . . . .	88
4.2.3	La chaîne de rendu . . . . .	90
4.2.4	Les possibilités de parallélisation des traitements . . . . .	90
4.2.5	Spécialisation du Z-buffer dans le cas de particules sphériques . . . . .	92
4.2.6	Conclusion sur le rendu . . . . .	95
<b>5</b>	<b>Conclusion</b>	<b>97</b>

<b>IV Applications dans le cadre des paysages naturels</b>	<b>99</b>
<b>1 Introduction</b>	<b>101</b>
<b>2 La végétation</b>	<b>103</b>
2.1 Introduction . . . . .	103
2.2 Principe . . . . .	103
2.2.1 Les attributs des particules et les caractéristiques des actions et fonctions . . . . .	105
2.2.1.1 Les attributs des particules . . . . .	105
2.2.1.2 États et actions . . . . .	106
2.2.1.3 Transitions et fonctions . . . . .	106
2.3 Premiers exemples . . . . .	108
2.4 Évolution et Influences externes . . . . .	110
2.4.1 La croissance . . . . .	110
2.4.2 Les contraintes spatiales . . . . .	111
2.4.3 Les contraintes spatiales et temporelles . . . . .	111
2.5 Rendu . . . . .	113
2.5.1 Fonction d'illumination probabiliste . . . . .	113
2.5.2 Formes et textures des particules . . . . .	114
2.6 Résultats . . . . .	115
2.7 Extensions . . . . .	118
2.8 Conclusion . . . . .	118
<b>3 Une autre application : les nuages</b>	<b>121</b>
3.1 Introduction . . . . .	121
3.2 Des particules et des automates pour les nuages . . . . .	121
3.3 Le rendu des nuages . . . . .	123
3.4 Conclusion . . . . .	124
<b>4 Conclusion</b>	<b>127</b>
<b>V Modélisation multi-échelle</b>	<b>129</b>
<b>1 Introduction</b>	<b>131</b>
<b>2 Niveaux de détail dans un paysage</b>	<b>133</b>
<b>3 Les objets multi-textures</b>	<b>137</b>
3.1 Le calcul des cartes . . . . .	137
3.2 La re-projection des cartes . . . . .	138
3.2.1 Construction de micro-polygones . . . . .	139



3.2.2	Tracer de rayon . . . . .	140
3.2.3	Re-projection directe . . . . .	140
3.3	Problème, amélioration et extensions . . . . .	142
3.4	Application aux éléments naturels . . . . .	144
3.4.1	Calcul des vues . . . . .	145
3.4.2	Le rendu de l'objet naturel . . . . .	145
3.5	Conclusion sur les objets multi-texture . . . . .	147
<b>4</b>	<b>Conclusion</b>	<b>149</b>
<b>VI</b>	<b>Un système pour la synthèse d'images de paysages</b>	<b>151</b>
<b>1</b>	<b>Introduction</b>	<b>153</b>
<b>2</b>	<b>L'architecture</b>	<b>155</b>
<b>3</b>	<b>Mise en œuvre</b>	<b>159</b>
<b>4</b>	<b>Résultats</b>	<b>161</b>
	La représentation et le rendu du terrain . . . . .	161
	Le calcul des ombres portées . . . . .	161
	Performances . . . . .	162
<b>VII</b>	<b>Conclusion</b>	<b>165</b>
<b>VIII</b>	<b>Planches couleurs</b>	<b>171</b>
<b>IX</b>	<b>Bibliographie</b>	<b>177</b>

Première partie

**Introduction**



## La motivation

Les raisons qui m'ont poussé à étudier le problème de la synthèse d'images de paysages naturels sont multiples. Il s'agit en premier lieu d'un intérêt personnel. Les paysages sont pour moi des images vivantes d'une grande beauté, dont on ne peut se lasser. Leur diversité et leur richesse sont le résultat d'une évolution dont la complexité nous dépasse mais dont nous sommes à la fois responsables et dépendants.

La complexité est le problème central quand on tente de modéliser des phénomènes naturels. C'est pourquoi la synthèse d'images réalistes<sup>1</sup> de paysages naturels reste encore un challenge en infographie. Alors que des méthodes permettent aujourd'hui de simuler précisément certains environnements pour obtenir des images fixes ou animées d'un réalisme presque parfait, les éléments naturels semblent échapper encore à ces tentatives de réduction mathématique et algorithmique. C'est donc un sujet encore ouvert dans un domaine de plus en plus difficile.

Une autre raison qui m'a incité à choisir un tel sujet de thèse est l'aspect « pluridisciplinaire » du problème. Il ne s'agissait pas d'approfondir une technique complexe et difficile de rendu ou de modélisation, mais plutôt de faire une synthèse et une adaptation d'approches existantes pour résoudre un problème spécifique. Ainsi, j'ai abordé dans ce travail de nombreux aspects de la synthèse d'images, que ce soit au niveau de la modélisation des objets ou de leur rendu.

La synthèse d'images réalistes de paysages naturels peut trouver des applications dans plusieurs domaines. Il existe en effet une demande pour de tels outils, capables de fournir une visualisation convainquante pour des non-spécialistes d'un environnement naturel. On peut citer en premier lieu, les études d'impact en milieu naturel dans des projets d'aménagement du territoire. Les simulateurs de pilotage ont également besoin de techniques pour la visualisation réaliste des paysages. Enfin, le marché important des effets cinématographiques et vidéos a de plus en plus besoin de l'image de synthèse. Il s'agit en particulier de représenter des environnements réels ou imaginaires, souvent trop complexes et trop difficiles à repro-

---

1. Par réalisme, nous caractérisons la faculté de susciter chez un observateur les mêmes sentiments que s'il regardait une photographie ou un film d'un paysage réel. C'est à dire les mêmes impressions que devant **une représentation de la réalité.**

duire par les techniques d'effets spéciaux classiques.

Finalement, un des intérêt, et non des moindres, de ce sujet est la facilité à communiquer le résultat. Toute personne est capable de donner son jugement sur la qualité d'un paysage, c'est à dire non seulement de le trouver beau ou laid, mais aussi d'apprécier le réalisme de l'image, comme le vent dans les arbres ou la forme des nuages. J'ai donc la satisfaction de pouvoir montrer le résultat de mes recherches à toute personne et d'avoir un avis intéressant en retour.

Ainsi, je pense que le problème de la synthèse d'images réalistes de paysages est une des dernières quêtes qu'il reste à accomplir dans le domaine de l'infographie. Cet objectif marquera une conjonction entre l'artificiel et le naturel.

## Les objectifs

Notre objectif est de spécifier et de réaliser un système pour la synthèse d'images de paysages naturels. C'est un projet ambitieux. De nombreux aspects de l'informatique graphique seront abordés et des choix devront être faits. Définissons tout d'abord ce que doit être un système de synthèse de paysages naturels :

- il doit permettre de modéliser et de rendre les objets et les phénomènes naturels qui composent un paysage : la végétation, le sol, l'eau sous toutes ses formes, l'atmosphère et le ciel, mais aussi les conditions météorologiques.
- il doit, d'une part, privilégier le réalisme des images obtenues, et d'autre part, avoir des temps de calcul raisonnables pour permettre la production d'animations.

La synthèse d'images de paysages naturels reste encore un problème peu abordé en infographie. Si des images convaincantes de végétaux, de reliefs et de nuages ont été produites, peu de scènes de paysages intégrant l'ensemble de ces éléments ont été présentées. La complexité à toutes les échelles de ces structures naturelles défie les techniques traditionnelles de modélisation et de rendu de scènes 3D. Pourtant, la demande de telles simulations est importante, que ce soit dans le domaine de la prospection et des études d'impact ou dans celui des effets spéciaux cinématographiques.

Quelles sont les spécificités de ce problème qui le rendent si difficile?

- une grande complexité géométrique **à toutes les échelles**. Que ce soit au niveau de la feuille ou de la fleur, jusqu'au relief des montagnes, aucun élément ne présente la régularité des objets artificiels;
- un caractère dynamique complexe (comme par exemple la croissance des végétaux ou le vent dans les branches d'un arbre);

- l’interdépendance des éléments naturels (la relation entre le sol et la végétation, l’influence mutuelle d’un arbre sur ses voisins);
- la difficulté de dénombrer et de classer les éléments composants un paysage.

Il n’est donc pas possible aujourd’hui de réaliser un tel système et envisager qu’il passe avec succès le “test de Turing” : c’est à dire que les images de synthèse produites soient indifférenciables d’images réelles de paysages.

Je définirai un certain nombre d’objectifs qui sont, à mon échelle, plus abordables :

- le système doit permettre d’intégrer des éléments de végétation, mais sans prétendre couvrir la diversité de la nature;
- des éléments indispensables comme le relief du terrain, les nuages ou les rivières doivent être représentés;
- seul un nombre restreint de phénomènes naturels sont intégrés, comme l’ensoleillement ou le vent;
- de ce fait, une caractéristique importante du système doit être la possibilité d’étendre a posteriori la palette des objets et des phénomènes couverts;
- le réalisme étant un des objectifs principaux, le rendu des éléments naturels doit être précis et intégrer l’ombrage, la texture des surfaces ...

Les solutions envisagées pour résoudre ces problèmes sont approchées sous l’angle de la synthèse d’images. Les mécanismes de modélisation sont ainsi basés sur la description géométrique plutôt que sur la simulation des forces ou des processus qui ont engendré l’élément. D’autre part, les techniques choisies pour modéliser et rendre les différents éléments naturels doivent répondre aux contraintes très spécifiques d’un système de rendu de paysages. L’architecture interne de ce système est également un point important. Celle-ci doit gérer efficacement un volume de données important, coordonner la modélisation et le rendu d’éléments indépendants, tout en garantissant la cohérence de la scène. L’optimisation des performances, en terme de temps de calcul, est aussi un critère qui guidera la conception de ce système.

Ainsi, nous fixons les critères suivants qui gouvernent nos choix quant aux méthodes de modélisation et de rendu :

- la modélisation doit s’appuyer sur des paramètres concrets de la morphologie des objets. Un non-spécialiste doit pouvoir créer facilement de nouveaux objets. Par exemple, la définition d’un végétal se fera à partir de l’observation de sa structure, indépendamment des mécanismes de croissance qui l’ont engendré.

- la modélisation doit permettre d'intégrer des paramètres externes à l'objet, que ce soit des influences globales (vents, ensoleillement) ou locales (obstacles, tailles et traumatismes).
- la modélisation et le rendu doivent être des processus simultanés. Ainsi, il n'y a pas de description géométrique à stocker, ni de représentation intermédiaire entre la phase de modélisation et celle de rendu.
- le rendu doit être réaliste, intégrant l'illumination, les textures et l'ombrage tout en restant efficace.
- le processus complet de modélisation/rendu d'un objet doit être rapide, afin d'intégrer en un temps raisonnable un nombre important d'éléments naturels dans une scène de paysage.

L'architecture du système doit prendre en compte les spécificités du problème, ce qui impose un certain nombre de contraintes, mais permet aussi quelques optimisations :

- dans un paysage, non seulement tous les éléments ne sont pas visibles, mais surtout, tous les éléments n'apparaissent pas avec le même niveau de détail sur l'image finale. La mise en œuvre et l'intégration de techniques multi-échelles est donc un point important de ce système.
- un nombre important d'éléments naturels sont pris en compte et, en général, ils peuvent être traités indépendamment. Ainsi, une distribution des calculs doit être possible sur une architecture parallèle.
- il doit être possible d'étendre facilement les fonctionnalités de ce système, en particulier en définissant de nouveaux éléments.
- les temps de calcul pouvant être longs, dans le cas d'animations par exemple, une interface conviviale doit permettre à l'utilisateur de connaître l'état du système et l'avancement des traitements.

## L'organisation du rapport

Nous présenterons tout d'abord un tour d'horizon détaillé des travaux relatifs à la synthèse des éléments naturels. Le chapitre le plus important concerne la création des végétaux, problème qui intéresse depuis longtemps les chercheurs en infographie. Un autre sujet important sera la modélisation et le rendu des reliefs naturels. Nous verrons les méthodes employées pour générer des terrains réalistes et les techniques utilisées pour les rendre. Enfin, des problèmes un peu moins abordés en synthèse d'images comme la simulation des phénomènes atmosphériques seront présentés à la fin de cette partie de l'état de l'art. Le chapitre suivant étudiera les possibilités des techniques de rendu et en particulier de celle

que nous avons retenue pour notre système, à savoir le Z-buffer. Nous verrons quelles sont les améliorations possibles de cette méthode permettant un rendu de qualité et des performances attractives.

Nous aborderons ensuite la partie de ce rapport présentant l'apport de cette thèse au domaine de l'infographie. Les notions de systèmes de particules et d'automates de contrôle seront alors définies. Un chapitre détaillé parlera de l'implémentation d'un tel système, que ce soit au niveau de la représentation de l'automate ou du rendu des particules.

L'application aux éléments naturels sera présentée dans la quatrième partie. Nous verrons comment le couple « système de particules - automate de contrôle » peut être utilisé pour créer des végétaux convaincants. Des exemples et des résultats illustreront la validité de cette approche. La modélisation et le rendu de nuages est également possible avec le même formalisme. Un chapitre sera consacré à ce sujet.

Comme nous l'avons remarqué dans la section précédente, l'aspect multi-échelle dans un système de rendu de paysages est important. La cinquième partie de ce rapport définira le problème et proposera des solutions adaptées aux éléments naturels et en particulier, à la végétation.

Finalement, nous présenterons l'architecture générale du système que nous avons développé. Nous montrerons qu'elle est particulièrement bien adaptée dans le cas de scènes très complexes telles qu'un environnement naturel complet.

Une conclusion récapitulera les avancés que nous avons proposées en les replaçant dans le domaine. Les développements futurs seront présentés avec une série d'objectifs à court et à long terme.





Deuxième partie

État de l'art



Spécifier et développer un système pour la synthèse d'images de paysages naturels nécessite une revue importante des travaux dans plusieurs domaines de l'infographie. Ainsi, cet état de l'art sera divisé en deux parties principales: les travaux traitant de la modélisation des éléments naturels et les travaux portant sur les techniques de rendu. Dans un premier temps nous proposons un tour d'horizon des méthodes pour la modélisation des objets et des phénomènes naturels avec une section importante consacrée à la création de végétaux. Deux autres aspects d'un paysage seront abordés: la modélisation et le rendu de terrain et les phénomènes atmosphériques. La deuxième grande partie de cet état de l'art traitera des méthodes pour le rendu de scènes 3D, en particulier des techniques adaptées aux contraintes d'un système de synthèse d'images de paysages.

L'étendue des travaux couverts dans cet état de l'art est donc très grande et certains choix ont dû être faits afin d'en limiter le volume. Les critères qui ont guidé ces choix ont été dictés par les objectifs et les limitations particulières fixés en introduction (cf. Introduction - Les objectifs).

Pour finir, nous résumerons les apports de cette revue et rappellerons les caractéristiques des travaux les plus intéressants qui ont inspiré la réalisation de notre système.



# Chapitre 1

## Les objets et les phénomènes naturels

Deux classes d'éléments peuvent intervenir dans la synthèse de paysages naturels : les phénomènes et les objets naturels. On distingue phénomènes et objets de la façon suivante :

- les **objets naturels** sont solides et concrets. Ils interagissent avec la lumière selon les modèles de réflexion matière-lumière. Dans le cadre de la synthèse de paysages, ce sont les terrains et les végétaux.
- les **phénomènes naturels** ne sont pas matériels mais agissent sur la lumière. Ils représentent la classe des éléments qui ne sont pas tangibles ou dont la surface n'est pas nette comme l'eau, le brouillard et tous les phénomènes atmosphériques. En général, les phénomènes naturels n'ont pas de géométrie bien définie et se manifestent par la transformation qu'ils imposent à la lumière (bien que les objets transforment eux aussi la lumière par réflexion, c'est d'ailleurs la raison pour laquelle nous les voyons, ils ont cependant un caractère tangible qui les différencie des phénomènes).

Les modèles qui ont été proposés pour la modélisation et le rendu d'objets naturels sont très variés et très différents. Dans cet état de l'art, nous ne prétendons pas couvrir la totalité des travaux portant sur ces sujets mais nous proposons une sélection de publications qui couvre l'éventail des méthodes existantes.

La première et la plus importante section de cet état de l'art concerne la synthèse des végétaux. Nous présenterons ensuite un tour d'horizon des travaux portant sur la modélisation et le rendu des reliefs. La troisième section parlera des méthodes pour prendre en compte les phénomènes atmosphériques dans une scène naturelle.

### 1.1 Les végétaux

Les végétaux sont des éléments très importants des paysages. Rares sont les scènes où ils n'apparaissent pas. Ce peut être seulement de l'herbe ou du gazon, mais aussi des buissons

ou des arbustes et bien sûr des arbres. Depuis plus de vingt ans, les chercheurs ont proposé des modèles pour représenter les végétaux. Et pourtant, encore aujourd'hui, aucun de ces travaux ne propose un modèle capable de reproduire rapidement, avec un parfait réalisme, la diversité des végétaux d'un paysage.

Les modèles pour la synthèse d'images de végétaux se divisent essentiellement en deux grandes catégories :

- **les modèles structurels** proposent dans un premier temps de construire la structure de l'objet qui est ensuite interprétée pour produire une description géométrique 3D. Un procédé de rendu est finalement chargé de calculer l'image de cette description en fonction des paramètres de l'environnement (position de l'observateur, données sur l'éclairage ...).
- **les modèles « impressionnistes »** mettent l'accent sur l'impact visuel du végétal, en général, au détriment de sa complexité structurale. Le processus de modélisation et de rendu sont en général confondus.

Dans cet état de l'art, nous présentons un certain nombre de modèles appartenant à ces deux catégories. Nous évaluerons ensuite leurs caractéristiques en fonction de certains critères importants pour leur intégration dans un système de rendu de paysages. Ces critères sont les suivants :

- **le réalisme des images** : le modèle doit permettre de produire des images réalistes de végétaux. C'est à dire que l'impression visuelle doit être aussi proche que possible de celle produite par l'objet réel.
- **la diversité des espèces** : plusieurs espèces différentes de végétaux doivent pouvoir être générées par le modèle. Plus l'éventail des espèces possibles sera grand, plus le modèle sera intéressant.
- **la nature des données nécessaires à la description** : les connaissances requises pour la description d'un végétal peuvent être très différentes d'un modèle à l'autre. Elles peuvent être de nature botanique, purement géométrique ou totalement empirique.
- **la complexité et les temps des calculs** : il est important que le modèle puisse produire rapidement la structure du végétal et éventuellement son rendu (dans le cas d'une méthode de rendu spécifique comme c'est le cas pour les modèles "impressionnistes"). En effet, une scène de paysage naturel peut comporter un grand nombre de ces éléments.
- **les influences externes** : le modèle doit pouvoir intégrer les influences d'un certain nombre de phénomènes externes (paramètres exogènes). Ce sont par exemple les effets du soleil, du vent, du type de terrain ou du voisinage.

Il s'agira donc, pour chacun des modèles présentés, d'évaluer ces différents critères. Cette classification nous donnera des indications quant aux caractéristiques requises pour qu'un modèle de synthèse des végétaux réponde aux problèmes spécifiques du rendu de paysages naturels.

### 1.1.1 Les modèles structurels

Cette première approche paraît la plus évidente : elle consiste à construire par une méthode mathématique ou algorithmique la structure ramifiée du végétal. En effet, le réseau ramifié sous-jacent d'un végétal est en général caractéristique de son espèce. Ensuite, l'interprétation géométrique et le rendu de cette topologie devront produire l'image voulue. Ces dernières étapes sont très importantes car le réalisme en dépend pour une grande part. Dans la suite, nous présentons par ordre chronologique un échantillon représentatif des méthodes structurelles pour la modélisation des végétaux.

#### 1.1.1.1 Qu'est ce que la structure ramifiée d'un végétal ?

La structure d'un végétal résulte essentiellement de la croissance de ses axes : les bourgeons qui apparaissent sur ces axes peuvent, soit continuer la croissance de l'axe soit donner naissance à une feuille, soit enfin créer un nouvel axe (c'est la ramification).

On distingue essentiellement trois types de ramifications :

- **la ramification continue** : dans ce cas, chaque bourgeon donne naissance à un nouvel axe latéral (cf. figure 1.1.a);
- **la ramification rythmique** : seuls certains bourgeons engendrent régulièrement de nouveaux axes (cf. figure 1.1.b);
- **la ramification diffuse** : seuls certains bourgeons aléatoirement répartis donnent naissance à un axe (cf. figure 1.1.c).

De plus, pour une espèce donnée de végétaux, le type de ramification est fonction du niveau de l'axe dans la structure ramifiée.

Le développement de chaque axe est également une caractéristique des végétaux, il peut être de deux types :

- **la croissance orthotropique** : la tendance de développement de l'axe est verticale (cf. figure 1.2.a);
- **la croissance plagiotropique** : la tendance de développement de l'axe est horizontale (cf. figure 1.2.b).

En plus de ces aspects portant sur la ramification et la croissance, d'autres phénomènes sont observés et résultent de la mort des bourgeons ou de la taille du végétal. On



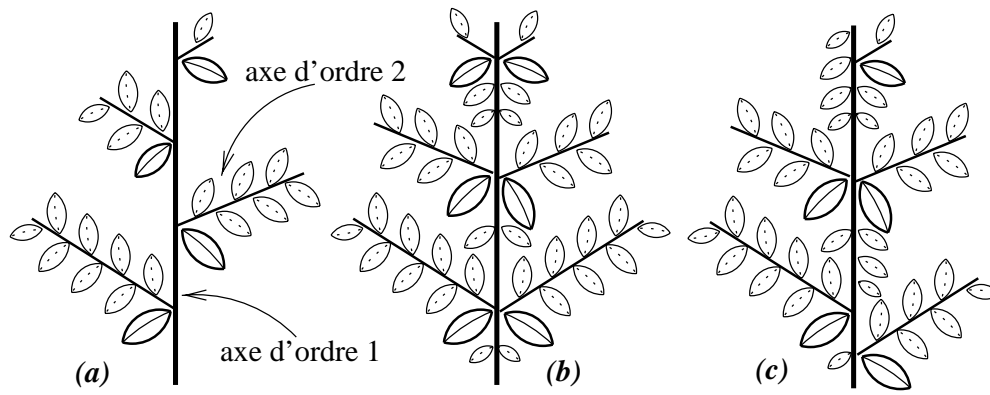


FIG. 1.1 – Les types de ramifications : (a) ramification continue, (b) ramification rythmique, (c) ramification diffuse.

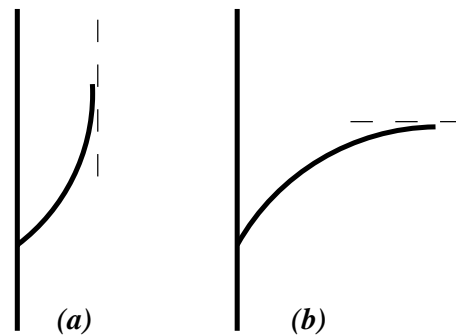


FIG. 1.2 – Tendances de croissance : (a) orthotropique, (b) plagiotropique.

peut ainsi observer un mode de croissance particulier appelé *sympodial*. Par opposition au développement *monopodial*, la croissance sympodiale est caractéristique d'une répétition des bourgeons latéraux (déclenchée par la mort du bourgeon apical ou par la taille de l'axe).

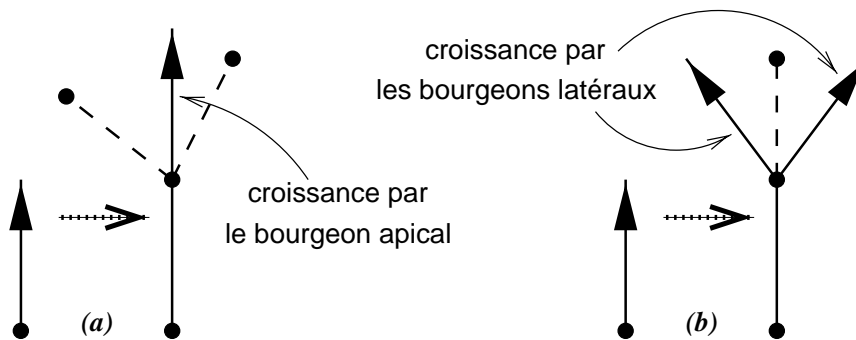


FIG. 1.3 – Mode de croissance : (a) monopodiale, (b) sympodiale.

En se basant sur ces caractéristiques, les botanistes ont proposé à partir de l'observation, une classification des arbres en modèles architecturaux. Cette classification a été utilisée par de Reffye [dREF<sup>+</sup>88] dans un modèle de développement des arbres (cf. 1.1.1.4, page 23).

### 1.1.1.2 Les premiers modèles à caractère géométrique [AK84]

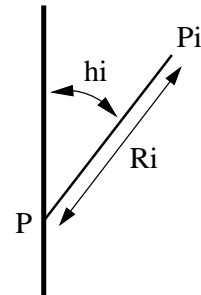
Les premiers chercheurs à s'intéresser à la modélisation informatique des végétaux furent les japonais Honda [Hon71], Aono et Kunii [AK84] ainsi que Kawaguchi [Kaw82]. Leurs approches sont essentiellement géométriques : les caractéristiques métriques de l'arborescence sont décrites par un certain nombre de paramètres géométriques comme les angles d'embranchement ou les rapports des longueurs des branches.

Aono et Kunii [AK84] proposent de nouveaux modèles géométriques pour calculer la direction et les caractéristiques des ramifications. Les paramètres des formules mathématiques sont déduits directement de l'observation de la morphologie d'arbres réels. Leurs modèles sont peu à peu étendus pour prendre en compte l'effet d'influences externes comme le vent, le soleil ou la gravité. De plus ils permettent des ramifications de type binaire et ternaire (à chaque ramification, deux ou trois branches filles sont créées) et des variations complexes des angles de branchement (en fonction de la distance au sol par exemple).

Leurs modèles déduisent la position 3D d'une branche fille à partir de formules de type :

$$P_i = P + R_i * f(h_i)$$

où  $P_i$  représente la position de la  $i^e$  branche fille,  $P$  la position de la branche mère,  $R_i$  le facteur de contraction de la branche fille par rapport à la branche mère et  $h_i$  les angles de branchement.



Pour prendre en compte les influences externes, ils introduisent un terme supplémentaire appelé déviation uniforme qui est appliquée à la position de chaque branche fille. De plus, pour prendre en compte la variation des paramètres de branchement par rapport au niveau de ramification, ils font évoluer les angles de branchement dans un intervalle en fonction du niveau de croissance.

FIG. 1.4 – Les paramètres de ramification.

Ces modèles ont été regroupés dans un système interactif appelé *A-system*. La visualisation des objets construits est simplement réalisée par des segments de droite d'épaisseur variable.

Cette première approche peut aussi être classée dans la catégorie des modèles empiriques (au sens de la taxinomie proposée par Fournier pour la modélisation des éléments

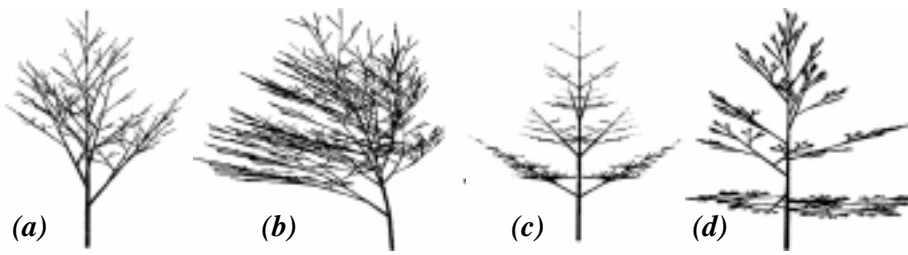


FIG. 1.5 – *Les apports des différents modèles de Aono : (a) modèle de base GMT1, (b) modèle GMT2 avec déviation, (c) modèle GMT3 avec ramification ternaire, (d) modèle GMT4 avec variation des paramètres de branchement en fonction du niveau de croissance [AK84].*

naturels [Fou89]). En effet, la paramétrisation du modèle géométrique est possible grâce à des mesures réalisées sur des végétaux et le modèle lui-même est déduit de l'observation. Cependant, il est probable que plus le nombre de végétaux couverts est grand, plus ces modèles géométriques doivent être complexes. On peut noter que cette première approche a beaucoup de points communs avec les travaux récents de Weber et Penn [WP95] (cf. section 1.1.1.7, page 33).

### 1.1.1.3 Les modèles fractals [Opp86, Smi84]

Une des caractéristiques plaisante des modèles fractals est la grande capacité d'amplification des données : des modèles à l'apparence très complexe peuvent être générés à partir d'un faible ensemble de paramètres. Cependant, si cette approche peut paraître a priori intéressante pour la modélisation des végétaux, les modèles fractals sont somme toute assez peu utilisés dans ce contexte. En effet, si les critères de régularité et l'autosimilarité présents dans les modèles fractals existent aussi dans les éléments naturels, les fractals purement déterministes ne se prêtent pas à la synthèse de végétaux. En général, des facteurs stochastiques sont intégrés dans le processus de création pour briser la régularité de ces modèles.

Oppenheimer [Opp86] propose un modèle fractal pour générer des images de plantes et de structures non-organiques comme les deltas des fleuves ou les flocons de neige. Il implémente ce modèle par un programme récursif qui, à chaque itération, associe au nœud courant un certain nombre de sous-arbres transformés par une application linéaire (une simple matrice  $3 \times 3$ ). À un certain niveau de récursion, le processus de ramification s'arrête et une feuille est créée. Les paramètres géométriques de transformation des branches filles sont semblables à ceux utilisés par Aono et Kunii [AK84] (rotation et changement d'échelle), avec en plus, une information sur la forme des branches entre les nœuds de ramification (trajectoire rectiligne, spirale ou hélicoïdale). Pour permettre la diversité et briser l'autosimilarité stricte de ce modèle, Oppenheimer spécifie pour chacun des paramètres, une valeur moyenne et un écart type. Pour cet auteur, ces facteurs contrôlant la topologie

et la géométrie sont semblables à des paramètres génétiques : ils caractérisent la forme et l'évolution des espèces générées.

Pour l'interprétation géométrique et le rendu des structures produites, Oppenheimer utilise des prismes polygonaux avec une texture d'écorce pour les branches les plus importantes. Cette texture est créée par *bump mapping* (perturbation de la normale en chaque point de la surface) à partir d'une fonction de bruit fractal.

D'autres approches fractales existent, comme les systèmes de fonctions itérées (IFS) [Ric95]. Les IFS ont de nombreuses applications dans le domaine de l'image et permettent en particulier de générer des objets à l'apparence naturelle. Les IFS sont des ensembles de fonctions affines contractantes qui convergent vers un unique point fixe. Les images générées ont donc un fort caractère d'autosimilarité et elles sont généralement utilisées pour représenter des feuilles ou des fougères.

Un autre auteur a envisagé la modélisation et la croissance des plantes comme un processus fractal. Smith [Smi84] définit une nouvelle classe d'objets appelée *graftals* issus de la théorie des langages formels. Bien que ce modèle partage un certain nombre de propriétés avec les objets fractals, Smith remarque qu'il ne fait pas strictement partie de cet ensemble.

Dans la mise en œuvre des méthodes à base de fonctions fractales, il y a généralement une séparation entre la modélisation d'une topologie et l'interprétation de cette structure pour produire une description géométrique de l'objet. Un grande partie du réalisme des images tient à la qualité de cette interprétation. Ainsi, certains auteurs comme Bloomenthal [Blo85], se sont intéressés à des problèmes tels la géométrie des branchements et des racines ou l'apparence de l'écorce.

Il apparaît difficile avec l'outil mathématique que sont les courbes fractales, de générer un ensemble de modèles précis de végétaux. La diversité des espèces s'accommode mal de l'autosimilarité stricte des fonctions fractales et la structure d'un végétal ne peut pas être facilement décrite par le faible nombre de paramètres de ces outils. J'écarte donc les modèles fractals purs de mon étude sur la synthèse des végétaux.

Par contre, les techniques d'amplification fractales sont utilisées avec plus de succès pour la modélisation des reliefs (cf. 1.2.1, page 47).

#### 1.1.1.4 Les modèles à base de paramètres botaniques [dREF<sup>+</sup>88]

De Reffye, Edelin, Françon, Jaeger et Puech [dREF<sup>+</sup>88] ont proposés un modèle botanique pour la synthèse des végétaux. Cette approche physique de la modélisation des plantes a pour principe la simulation de l'activité des bourgeons (cf. 1.1.1.1 Structure ramifiée, page 19) sur une échelle de temps discrète.

À un instant donné, un bourgeon peut évoluer vers l'un des quatre états suivants :

- le bourgeon se transforme en fleur et meurt;
- le bourgeon suspend son activité pendant une unité d'horloge;
- le bourgeon devient un inter-nœud composé de feuilles, de nouveaux bourgeons latéraux et d'un bourgeon apical;
- le bourgeon meurt.

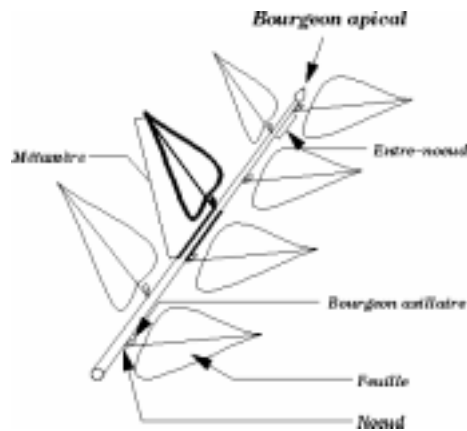


FIG. 1.6 – *Structure ramifiée et bourgeons.*

Des probabilités sont affectées à chacune des évolutions possibles du bourgeon et sont caractéristiques de l'espèce et du niveau de développement du végétal considéré. En plus de ces probabilités, ce modèle de croissance nécessite un certain nombre d'autres paramètres comme le nombre de bourgeons à chaque nœud ou encore les vitesses de croissance. Ces paramètres sont issus de connaissances précises en botanique et en morphologie des végétaux.

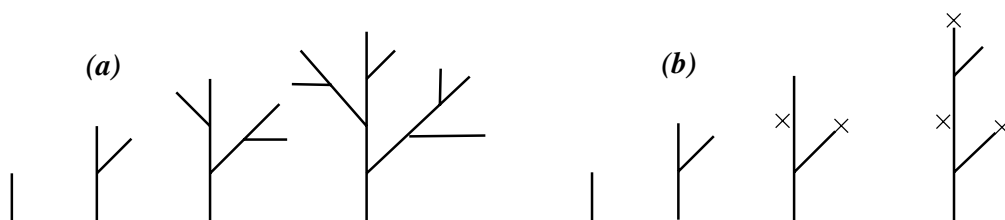


FIG. 1.7 – *Exemples de croissance : (a) les probabilités de pause et de mort des bourgeons sont nulles, (b) la probabilité de mort est non nulle.*

Des facteurs externes sont également facilement intégrés, comme par exemple la coupe des branches (tous les bourgeons d'une branche meurent) ou l'effet de la gravité sur les branches (en utilisant un paramètre d'élasticité dépendant de la nature du bois).

---

**Algorithme 1.1** Algorithme simplifié de simulation de croissance.
 

---

```

Pour chaque top d'horloge Faire
  Pour chaque bourgeon encore vivant Faire
    Si le bourgeon ne meurt pas Alors
      Si le bourgeon n'est pas en pause Alors
        Créer un inter-noeud
        Créer un bourgeon apical
        Pour chaque nouveau bourgeon possible Faire
          Si la ramification est possible
            Alors Créer le bourgeon latéral
        FinPour
      FinSi
    FinSi
  FinPour
FinPour
  
```

---

La description issue du moteur de croissance est ensuite interprétée comme un ensemble de primitives géométriques simples (tronc de cône pour les branches, polygones pour les feuilles) qui seront traitées par un système de rendu. Initialement, les auteurs ont utilisé un Z-buffer pour réaliser l'élimination des parties cachées et des méthodes simples d'illumination (éclairage Lambertien), cependant n'importe quel système de rendu peut être utilisé.

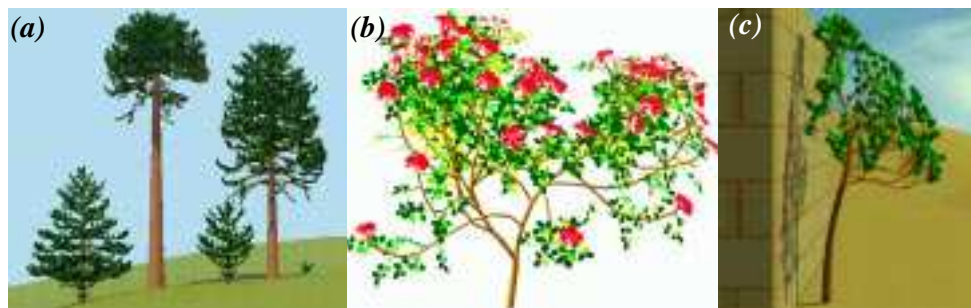


FIG. 1.8 – Résultats : (a) la croissance d'un *araucaria*, (b) un arbustes avec des fleurs, (c) un pin dont la croissance est influencée par le mur [dREF<sup>+</sup> 88] (voir aussi planches couleurs page 173).

Ce modèle très complet et très réaliste de simulation et de visualisation de la croissance des végétaux a abouti à un système de simulation commercialisé par le CIRAD (Centre de coopération Internationale en Recherche Agronomique pour le Développement) sous l'appellation AMAP. Il fonctionne sur stations de travail Silicon Graphics et s'interface avec des logiciels de synthèse d'images.

Cependant, dans le cadre de notre objectif de rendu de paysages, on peut relever un certain nombre d'inconvénients :

- des connaissances en botanique et morphologie des végétaux sont nécessaires pour paramétrer de nouveaux modèles. Ces paramètres peuvent être nombreux et difficiles à acquérir,
- les descriptions 3D produites peuvent être importantes (plusieurs dizaines de milliers de primitives pour des végétaux complexes comme les arbres). Un tel objet sera donc lourd à traiter pour un système de rendu,
- a priori, le moteur de croissance n'a pas la possibilité de dégrader facilement la structure produite et ainsi de pouvoir générer des modèles à différents niveaux de détail (pour une utilisation dans un système de rendu multi-échelle).

#### 1.1.1.5 Les modèles à base de grammaires [PLH88]

En 1968, Lindenmayer proposa un modèle de développement des structures biologiques basé sur un ensemble de règles de réécriture. Appelé L-system, ce modèle a été adapté à la génération des structures des végétaux. Une règle de réécriture est composée de deux membres : le prédécesseur et le successeur. À chaque itération et en parallèle, les occurrences du prédécesseur dans une chaîne sont remplacées par le successeur. Dans le cas des végétaux, cette chaîne représente la structure ramifiée de la plante (cf. figure 1.9).

Ce modèle simple présente l'avantage d'avoir des propriétés intéressantes par rapport au processus de croissance des végétaux. Par exemple, le principe de croissance apicale impose que seul le sommet des branches (apex) peut donner naissance à de nouvelles ramifications, les inter-nœuds ne pouvant que grandir.

Prusinkiewicz a beaucoup travaillé sur le modèle de Lindenmayer [PLH88, PHM93, Pru93, PJM94], et a produit de très belles images de plantes herbacées. Il met en évidence l'adéquation des méthodes à base de réécriture (grammaires hors contextes et grammaires contextuelles) aux mécanismes de croissance de végétaux (transfert des informations par les cellules ascendantes ou par les cellules voisines). L'accent est mis sur la simulation de la croissance. Des processus complexes de développement sont modélisés grâce à la possibilité des L-systems de propager des signaux entre la base et l'apex des structures ramifiées. Ainsi, les chaînes produites par ces grammaires peuvent représenter la topologie d'un vaste éventail de plantes, des plus simples au plus complexes.

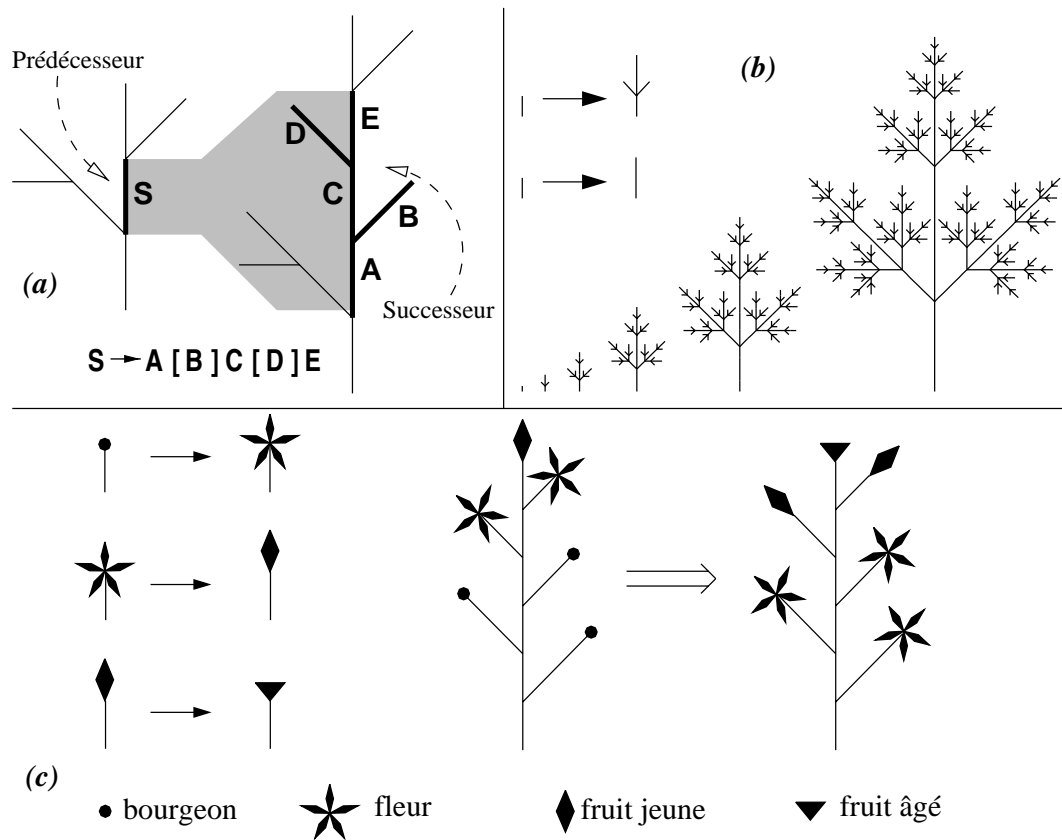


FIG. 1.9 – Principe de réécriture : (a) l'occurrence du prédécesseur est remplacé par la successeur, (b) développement d'une structure ramifiée à partir de deux règles de réécriture, (c) l'évolution des fleurs n'affecte pas la structure sous-jacente [PLH88].



FIG. 1.10 – Quelques exemples : (a) application des L-systems différentiels (DL-systems) [PHM93], la croissance est continue, (b) application des grammaires contextuelles, des signaux circulent pour activer un élément sous-jacent de la structure ramifiée.

Les facteurs qui influent sur la croissance d'un végétal sont de deux types :

- les mécanismes **endogènes** représentent l'action sur la croissance de paramètres internes à la plante, comme la propagation des informations entre les cellules;
- les mécanismes **exogènes** sont les influences externes à la plante, comme le voisinage ou la recherche de la lumière.



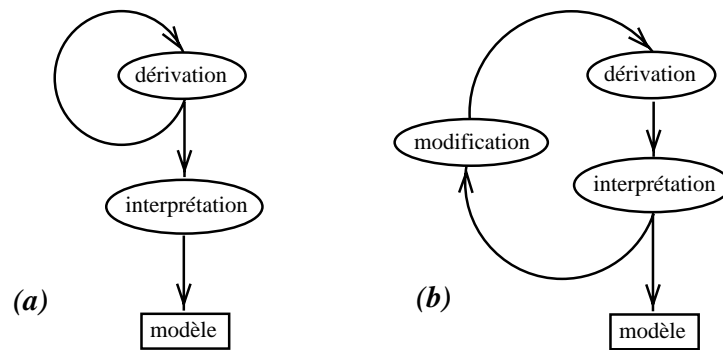


FIG. 1.11 – *Le processus de génération : (a) processus standard, chaque nouvelle itération crée un modèle, (b) « environmentally-sensitive L-system », à chaque itération les règles sont modifiées en fonction de critères géométriques.*

Les L-systems ne simulant que les mécanismes endogènes, Prusinkiewicz propose une extension qui intègre ces deux types de mécanismes [PJM94]. Les phases de génération de la topologie et d'interprétation géométrique étant distinctes dans les L-systems, une prise en compte de la position ou de l'orientation d'une branche durant sa croissance n'est pas possible. Dans le nouveau modèle appelé « environmentally-sensitive L-systems », l'interprétation géométrique est réalisée à chaque étape de production d'une nouvelle chaîne (cf. figure 1.11). Ainsi, les paramètres géométriques peuvent être intégrés dans les règles de production pour simuler par exemple la taille des branches et la réitération traumatique<sup>1</sup> (cf. figure 1.12).

---

1. **réitération traumatique** : à la suite d'une taille ou de la mort du bourgeon apical, un bourgeon du même axe jusqu'alors inactif peut donner naissance à une nouvelle branche.

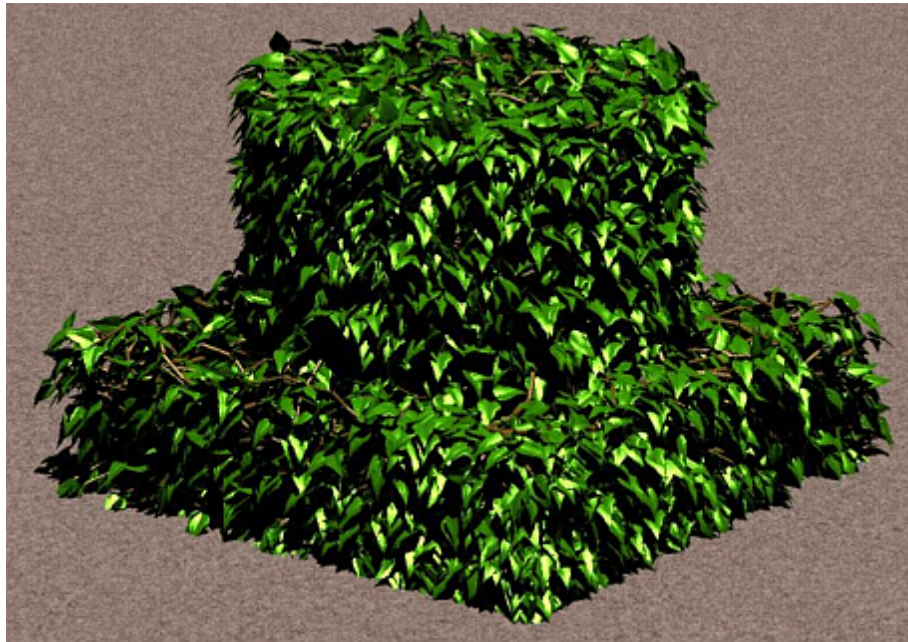


FIG. 1.12 – *Un exemple de croissance soumise à la taille. Il est ainsi possible avec cette classe de L-system de modéliser les plantes d'un jardin d'ornement [PJM94].*

La modélisation géométrique des topologies générées se fait simplement en interprétant les symboles de la chaîne par un système de type tortue graphique (cf. figure 1.13). Chaque symbole va, soit modifier la position courante de la tortue, soit appliquer une rotation à la direction courante, soit empiler/dépiler l'état courant du contexte graphique (dans le cas d'une ramification). Pour permettre une plus grande richesse et un contrôle accru des paramètres géométriques, Prusinkiewicz intègre dans l'alphabet des grammaires des symboles qui contrôlent l'interprétation graphique des topologies. Ce sont par exemple, la couleur, le diamètre ou la longueur des inter-nœuds.

Pour introduire un facteur aléatoire dans les L-systems, Prusinkiewicz ajoute une probabilité à chaque règle de production. Cette version stochastique des L-systems permet de produire plusieurs spécimens différents d'une même espèce de plante (cf. figure 1.14).

Les modèles à base de L-systems développés par Prusinkiewicz sont parmi les plus complets et les plus précis. Ils s'appliquent avec succès à de nombreuses espèces aussi bien au niveau du réalisme visuel, que de la dynamique de la croissance ou des influences externes. Le modèle s'avère facilement extensible et Prusinkiewicz [Pru93] remarque qu'il peut reproduire les modèles proposés par Aono [AK84] ou de Reffye [dREF<sup>+</sup>88].

Récemment, Lintermann [LD96] a proposé un système pour la modélisation de structures générées par L-systems. Le modèle est construit interactivement par assemblage de

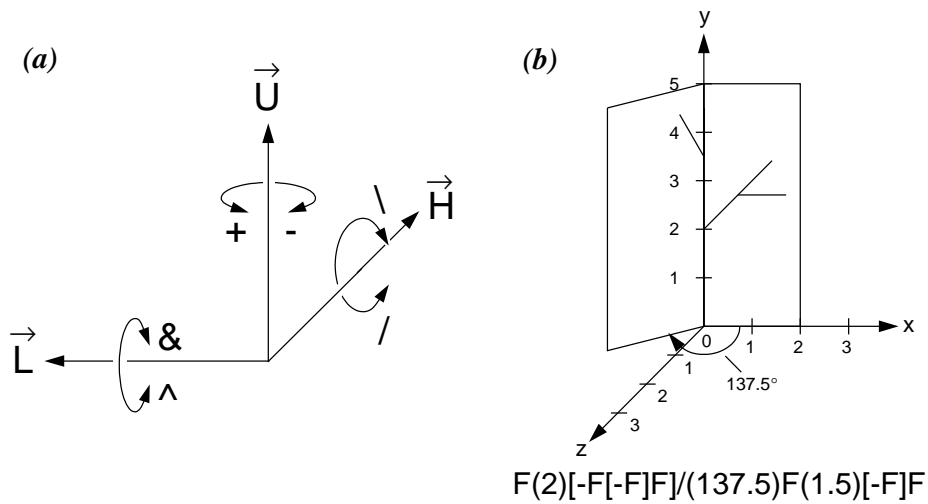


FIG. 1.13 – *Interprétation géométrique en trois dimensions : (a) des symboles de l’alphabet (+, -, &, /...) permettent de contrôler l’orientation, (b) un exemple de chaîne produite par L-system paramétrique et son interprétation géométrique.*



FIG. 1.14 – *Un champ de tournesols produit par L-systems stochastiques [PL90] (voir aussi planches couleurs page 173).*

composants représentant des fonctions ou des objets (cf. figure 1.15).

Cependant, certains aspects des L-systems rendent difficiles leur intégration dans notre système de rendu de paysage :

- la séparation des processus de modélisation topologique et d’interprétation géomé-

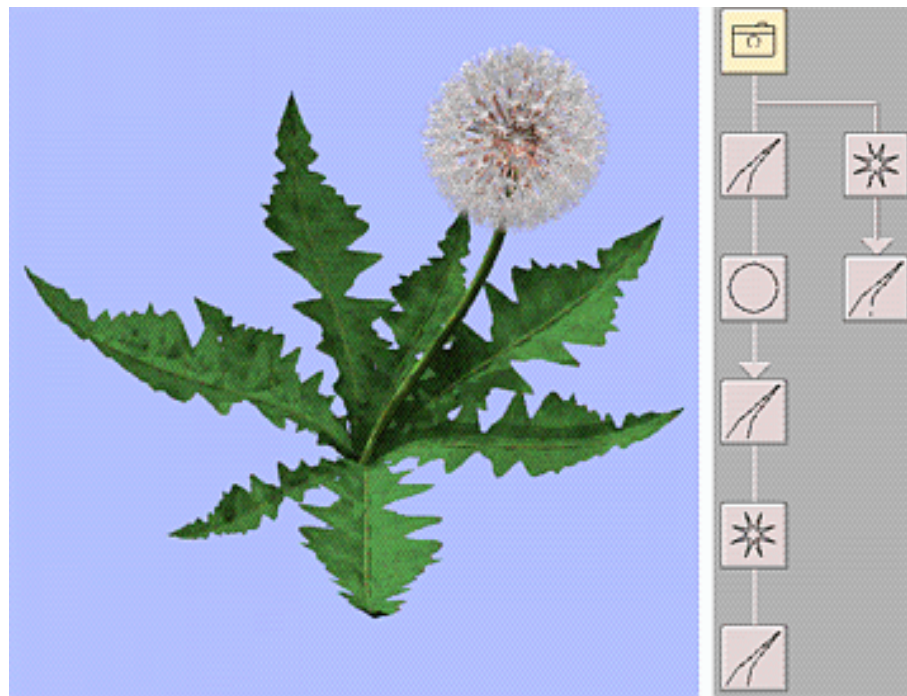


FIG. 1.15 – Modèle d'un pissenlit [LD96]. Les composants connectés définissent la construction de la tige et des feuilles ainsi que de la fleur. Ces composants représentent soit des primitives géométriques, soit des constructeurs.

trique. Comme pour de nombreux modèles « orientés structure », la modélisation d'un objet passe d'abord par la création d'une topologie (la structure ramifiée et ses paramètres) et ensuite par une conversion en primitives géométriques. Ces deux étapes distinctes imposent donc le stockage temporaire de cette structure (qui peut être très importante dans le cas d'un arbre par exemple).

- dans sa version la plus complète et la plus intéressante, le modèle de Prusinkiewicz devient assez complexe à implémenter.
- la description en terme de règles de production d'une espèce particulière de végétal n'est pas chose aisée. Une certaine pratique et des connaissances en botanique font que cette technique n'est pas facilement abordable. Cependant, l'outil proposé par Lintermann semble permettre une construction facile des végétaux.

#### 1.1.1.6 Les modèles combinatoires [VEJA89]

Le modèle développé par Viennot, Eyrolles, Janey et Arquès [VEJA89] est une approche originale de la modélisation des arbres. Selon cette méthode combinatoire, la topologie d'un arbre binaire est générée à partir d'une matrice de ramification. La matrice de ramification donne, pour chaque nœud d'ordre  $k$ , les probabilités de ramification vers un couple de

nœuds d'ordre  $(j, k)$  avec  $j < k$  ou d'ordre  $(k - 1, k - 1)$ .

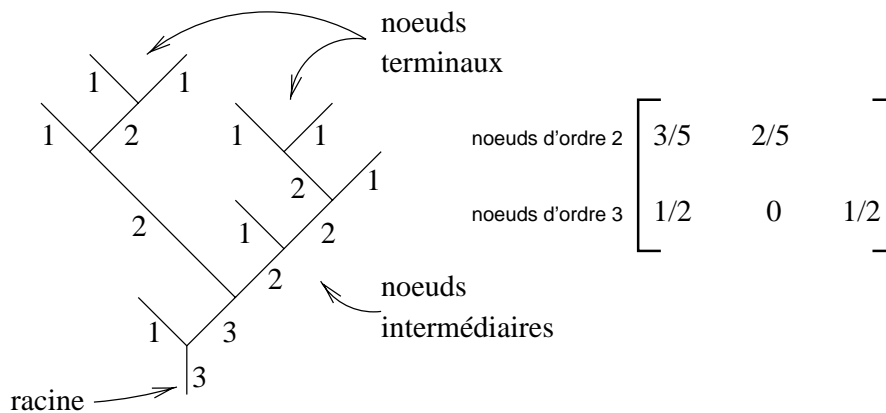


FIG. 1.16 – Un arbre binaire et la matrice de ramification associée.

La matrice traduit donc un certain nombre d'aspects visuels (arbre touffu, effilé, bien équilibré) caractéristiques des arbres. Un algorithme utilise cette matrice stochastique triangulaire pour générer des arbres aléatoires. Il est ainsi possible d'obtenir un ensemble d'arbres dont les caractéristiques topologiques sont directement déduites de l'analyse de la forme d'arbres réels.

Une fois la topologie de l'arbre modélisée, une interprétation géométrique assez rudimentaire va produire une description 3D de l'objet. Trois paramètres interviennent dans la géométrie: les angles entre les branches, l'épaisseur et la longueur d'une branche. Ils sont simplement définis comme fonction de l'ordre des branches.



FIG. 1.17 – Deux arbres générés par la méthode des matrices de ramification [VEJA89].

Ce même modèle est aussi utilisé pour générer la topologie de réseaux fluviaux, ou la

forme des feuilles.

Cette approche originale a cependant des inconvénients qui rendent son utilisation difficile: générer la topologie d'arbres binaires est loin d'être suffisant pour modéliser un arbre. La topologie de l'arbre sous-jacent à un végétal est certainement une caractéristique fondamentale de la structure ramifiée, mais de nombreux autres paramètres doivent être intégrés pour représenter la diversité et la complexité des espèces végétales.

#### 1.1.1.7 Les modèles récents à caractère géométrique [WP95]

Récemment, Weber et Penn [WP95] ont présenté un modèle intéressant pour la création d'arbres de synthèse réalistes. Leur approche est purement géométrique, mais grâce à un ensemble important de paramètres précis, les résultats sont convaincants. Les contraintes qui ont guidé la conception de ce modèle sont les suivantes :

- **réalisme** : l'aspect du feuillage et des branches doit être cohérent et visuellement satisfaisant pour un observateur mobile dans la scène;
- **richesse** : le modèle doit être capable de générer une grande variété d'espèces différentes;
- **aléatoire** : un grand nombre d'arbres différents d'une même espèce doivent pouvoir être créés;
- **facile** : l'utilisation de ce modèle doit être possible sans connaissances spécialisées;
- **compact** : la spécification d'un arbre doit consister en un ensemble réduit de données;
- **multi-échelle** : le modèle géométrique doit pouvoir être dégradé.

La spécification d'une nouvelle espèce d'arbres nécessite la définition d'un ensemble de paramètres géométriques. Weber et Penn remarquent qu'il existe rarement plus de quatre niveaux de récursion dans la structure ramifiée d'un arbre. Les paramètres géométriques généraux seront donc spécifiés pour chacun de ces niveaux de récursion. De plus, chaque paramètre est accompagné d'une valeur de tolérance qui fixe pour ce critère un intervalle de valeurs possibles. Les paramètres principaux (pour chaque niveau de récursion) sont les suivants (cf. figure 1.18) :

- *CurveRes* et *Curve* définissent le nombre et la rotation des troncs de cône qui composent une branche;
- *SegSplits* donne le nombre de « clones » créés sur une branche. Ces clones ont le même niveau de récursion que la branche et héritent des mêmes paramètres géométriques;
- *SplitAngle* définit l'angle de ramification des clones;
- *Branches* définit le nombre maximum de sous-branches que peut porter une branche;

- *Length* est le rapport de la longueur de la sous-branche par rapport à la longueur de la branche;
- *DownAngle* et *Rotate* définissent les angles de branchement d'une sous-branche.

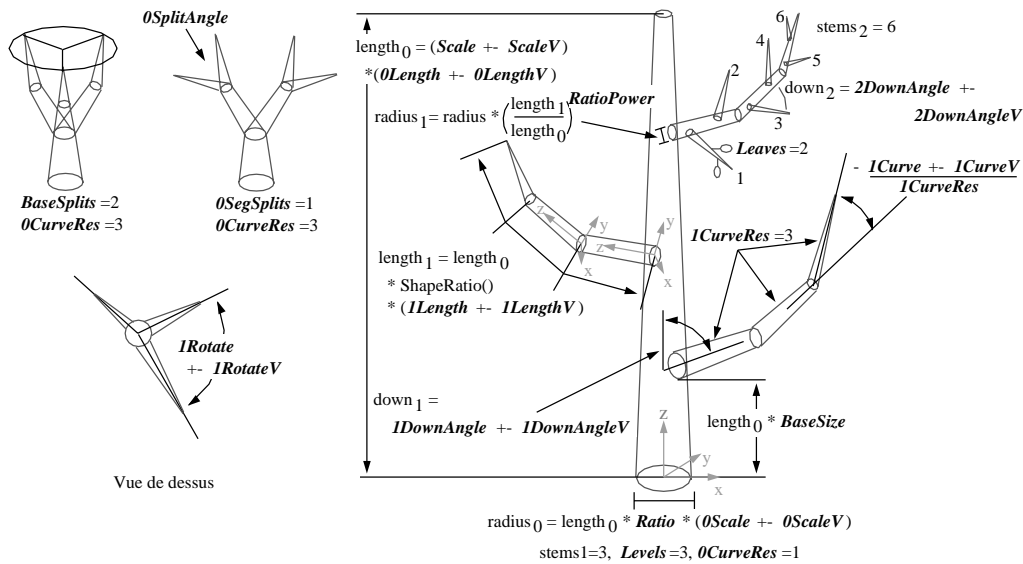


FIG. 1.18 – Les paramètres géométriques. Ces paramètres sont définis pour chaque niveau de ramification (par exemple  $0CurveRes$ ,  $1CurveRes$ ,  $2CurveRes$  ...). À certains paramètres sont associés des valeurs de perturbation (par exemple  $0LengthV$  pour  $0Length$ ) [WP95].

En plus de ces paramètres simples, de nombreuses autres valeurs sont utilisées pour la définition d'un arbre et font de ce modèle un outil riche et puissant. *Shape* définit la forme générale de l'arbre en fixant les longueurs et les angles des sous-branches. Les différentes valeurs de *Shape* permettent de choisir une forme parmi huit pré-définies. D'autres paramètres comme *Taper*, *Flare* ou *Lobe* agissent sur le rayon des branches et permettent de prendre en compte des spécificités (variation périodique du rayon, évasement à la base du tronc, section du tronc non circulaire ...).

Pour le feuillage, le paramètre *Leaves* est utilisé pour spécifier la densité des feuilles sur les branches, tandis que *LeafShape* contrôle la morphologie des feuilles.

Pour élargir encore l'application de ce modèle, en particulier la prise en compte d'influences externes, Weber et Penn ont intégré un contrôle plus complexe sur la longueur et les angles de ramification des sous-branches. Les phénomènes de taille des branches et d'influences de vent ou du soleil peuvent ainsi être intégrés.

Les auteurs remarquent qu'une description géométrique générée par ce modèle peut être composée de 5000 à 100000 polygones. Cependant, dans de nombreuses scènes où les objets sont loins de l'observateur, une telle précision n'est pas nécessaire. Aussi, Weber et Penn

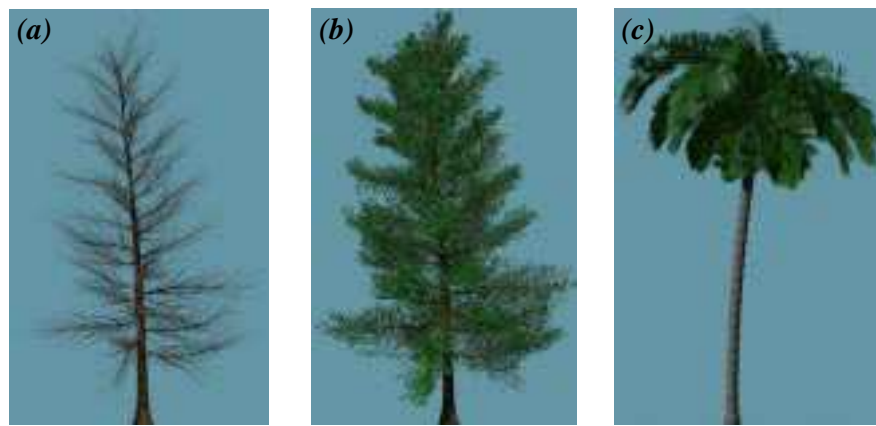


FIG. 1.19 – Quelques résultats: (a) un *Tupelo noir* (*nyssa sylvatica*) (b) le même *Tupelo* avec des feuilles, (c) un palmier [WP95] (voir aussi planches couleurs page 173).

proposent une interprétation adaptative du modèle au moment de produire la description géométrique. Ainsi, dans le cas d'un niveau de détail décroissant, les branches sont progressivement interprétées comme des lignes et les feuilles comme des points. Ensuite, toujours dans le cas d'un objet s'éloignant de l'observateur, certaines branches et certaines feuilles ne seront plus interprétées. Par exemple, un arbre proche de l'observateur composé de 100000 polygones sera rendu par 30 lignes et 1000 points à une distance de 2 kilomètres (cf. figure 1.20).



FIG. 1.20 – Modèles multi-échelles d'un tremble pour un rendu à 30, 60, 120, 240, 600 et 1200 mètres [WP95].

Ce modèle atteint les objectifs fixés par les auteurs : une grande variété d'espèces d'arbres peuvent être créées et ce, avec un très bon réalisme visuel. De plus, les connaissances requises pour la définition d'un nouveau modèle sont purement géométriques.

Bien que les auteurs introduisent une méthode de simplification du modèle dans le cas d'une application multi-échelle, le volume des primitives générées est encore trop important pour une utilisation dans un système temps réel ou dans une scène composée d'un très grand nombre d'éléments. La place mémoire requise pour stocker la géométrie d'un objet



complexe est également un obstacle important pour l'intégration d'un tel modèle dans un système de rendu de paysages.

### 1.1.2 Les modèles « impressionnistes »

#### 1.1.2.1 Qu'est ce qu'un modèle impressionniste?

Nous qualifions d'impressionnistes les modèles qui mettent l'accent sur la qualité de l'impression visuelle plutôt que sur la précision et le réalisme des structures produites. Ici, le critère est purement visuel, l'objectif étant atteint quand l'observateur qui regarde l'image de synthèse du végétal a « l'impression » de voir un élément réel. Ainsi, un tel modèle n'est pas forcément construit dans l'espace (3D), mais peut simplement produire une image convaincante (2D).

#### 1.1.2.2 Les modèles à base de textures [Gar84]

##### Textures 2D

Une technique simple pour visualiser des arbres est utilisée dans les systèmes de rendu temps réel (simulateurs, jeux vidéos). Elle consiste simplement à plaquer une texture représentant un arbre sur un polygone et veiller à ce que celui-ci soit toujours orienté en direction de l'observateur. Il n'est donc pas question ici de modèle permettant de générer des végétaux mais seulement d'une technique rudimentaire mais très efficace pour les visualiser. Les systèmes de rendu temps réel sont en général capables de rendre très rapidement les textures, ainsi un grand nombre de ces arbres peut être incorporé dans une scène.

Cependant, cette méthode a beaucoup d'inconvénients :

- il faut autant de textures que d'arbres différents que l'on veut intégrer à la scène. En général, le nombre de textures est limité par la taille de la mémoire disponible dans le système temps réel.
- il est très difficile d'incorporer des effets convaincants d'illumination et d'ombrage dans ce modèle.
- si l'observateur se déplace librement dans la scène, non seulement, il pourra remarquer que l'image de l'arbre est toujours la même, mais aussi qu'il est parfaitement « plat » en vue de dessus (sauf si des arrangements en croix des textures sont utilisés, cf. figure 1.21).

##### Textures 3D sur quadriques

Gardner [Gar84, Gar85] propose une méthode impressionniste pour la modélisation et la visualisation d'éléments naturels. Il s'intéresse en particulier à la création de nuages et d'arbres dans le contexte d'applications temps réel.

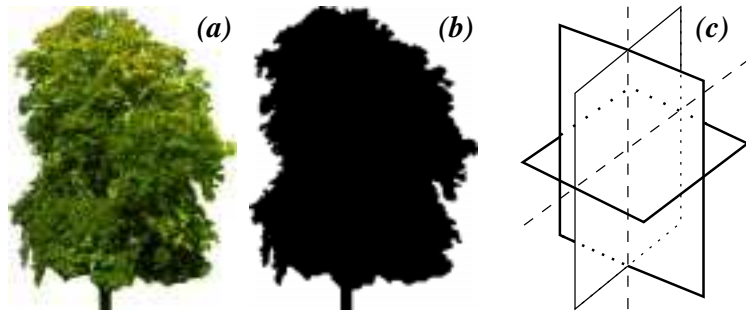


FIG. 1.21 – *Texture 2D pour la visualisation des arbres : (a) une texture d'un arbre réel, (B) le masque d'opacité associé, (c) l'arrangement en croix des textures pour une meilleure impression visuelle.*

Gardner introduit une nouvelle primitive appelée *ellipsoïde fractal*. La couleur et l'opacité des points de ces ellipsoïdes sont modulées par une texture fractale 3D. Un ou plusieurs de ces ellipsoïdes sont nécessaires pour modéliser un arbre ou un nuage.

La géométrie d'un ellipsoïde est simplement définie par l'équation  $(X - X_0)^2/R_x^2 + (Y - Y_0)^2/R_y^2 + (Z - Z_0)^2/R_z^2 = 1$ . Une texture fractale  $Fr(X, Y, Z)$  est calculée à partir du produit de séries de Fourier :

$$S_X = \sum_{i=1}^N C_i (\cos(\omega_{X_i}(X + \Phi_{G_X}) + \phi_{X_i}) + 1)$$

$$S_Y = \sum_{i=1}^N C_i (\cos(\omega_{Y_i}(Y + \Phi_{G_Y}) + \phi_{Y_i}) + 1)$$

$$S_Z = \sum_{i=1}^N C_i (\cos(\omega_{Z_i}(Z + \Phi_{G_Z}) + \phi_{Z_i}) + 1)$$

$$Fr(X, Y, Z) = S_X S_Y S_Z$$

Les paramètres  $C_i, \omega_{X_i}, \omega_{Y_i}, \omega_{Z_i}$  définissent le spectre des fréquences de la texture;  $\phi_{X_i}, \phi_{Y_i}, \phi_{Z_i}$  représentent les phases locales et sont utilisées pour introduire des variations aléatoires, tandis que  $\Phi_{X_i}, \Phi_{Y_i}, \Phi_{Z_i}$  sont les phases globales et permettent d'appliquer une translation à la texture.

Gardner utilise les ellipsoïdes fractals pour représenter des arbres. Ainsi, un seul ellipsoïde est utilisé pour simuler le feuillage. Dans le cas d'un modèle plus précis, un ellipsoïde par branche peut être nécessaire. Le tronc est également représenté à l'aide de cette primitive.

Cette technique simple et rapide peut aussi être utilisée pour visualiser des nuages ou de la fumée. Gardner utilise alors des ensembles d'ellipsoïdes fractals pour représenter ces

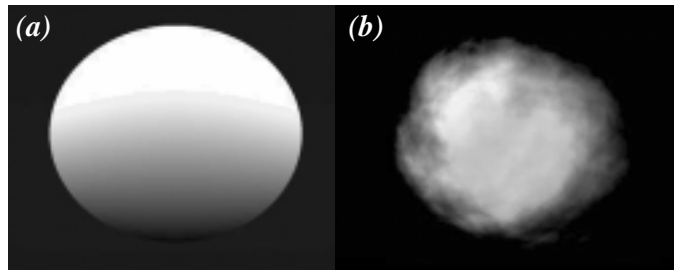


FIG. 1.22 – *Ellipsoïde fractal* : (a) un ellipsoïde sans texture, (b) la fonction  $Fr(X, Y, Z)$  est utilisée directement pour définir l'opacité de l'ellipsoïde [Gar84].

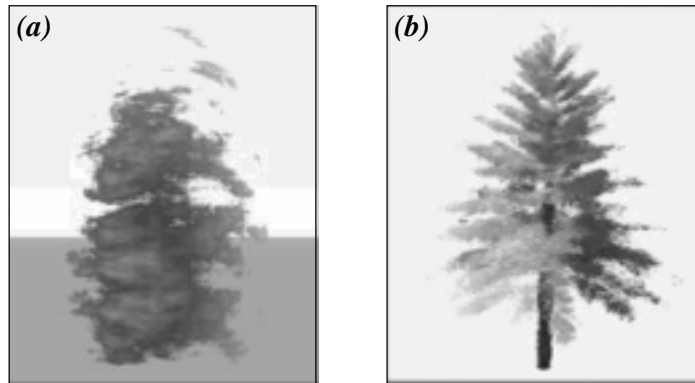


FIG. 1.23 – *Arbres à base d'ellipsoïde fractal* : (a) un seul ellipsoïde est utilisé pour représenter le feuillage de l'arbre, (b) plusieurs ellipsoïdes représentent la structure de l'arbre [Gar84].

phénomènes atmosphériques, statiques ou dynamiques.

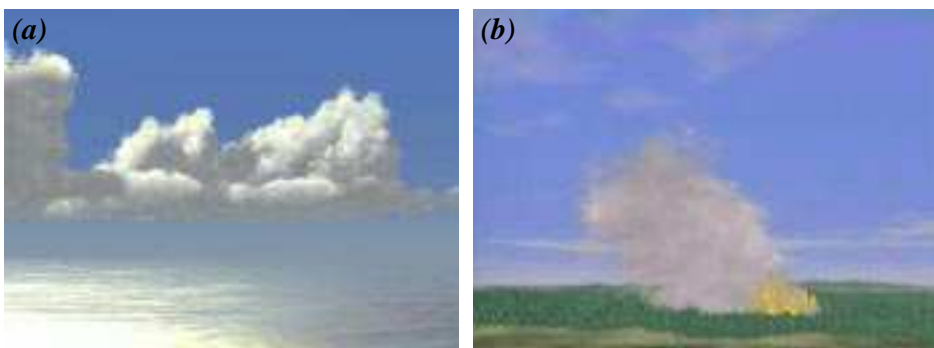


FIG. 1.24 – *Phénomènes atmosphériques* : (a) un nuage formé par un ensemble d'ellipsoïdes fractals, (b) simulation de fumée par animation d'ellipsoïdes fractals [Gar85].

Cette méthode permet une visualisation très rapide mais très peu détaillée des arbres.

Elle semble donc utilisable uniquement dans le cas d'éléments de second plan d'une scène naturelle. Il paraît en effet difficile de modéliser précisément une espèce particulière ou de représenter des détails comme les branches.

Dans un système de rendu de paysages utilisant une classification multi-échelle des éléments de la scène, les arbres éloignés de l'observateur pourraient être rendus efficacement par cette technique (cf. les objets multi-textures, chapitre 3, page 137).

### 1.1.2.3 Les modèles dans un espace de voxels [Gre89]

En 1989, Ned Greene a publié une nouvelle méthode pour modéliser les plantes dans un espace discret [Gre89]. Cette méthode met en œuvre un processus de croissance stochastique dans un espace de voxels<sup>2</sup>. Utiliser un espace de voxels permet de simplifier un certain nombre d'opérations :

- les tests d'intersection: chaque objet de nature différente étant représenté dans cet espace par les voxels qu'il occupe, un simple test suffit à déterminer si une position dans l'espace est occupée ou non.
- les mesures de proximité: en parcourant les voisins d'un voxel, on détermine rapidement les objets les plus proches.

Le modèle de simulation de Greene met l'accent sur la prise en compte des influences externes pouvant intervenir lors du développement d'un végétal. En effet, une plante réagit à son environnement, que ce soit en évitant les obstacles ou en recherchant la lumière. Peu de modèles structurels de végétaux intègrent de tels critères dans le processus de développement (les modèles développés par de Reffy [dREF<sup>+</sup>88] et Prusinkiewicz [PJM94] peuvent prendre en compte des facteurs exogènes). Greene propose donc une méthode basée sur la perception de l'environnement grâce aux propriétés intéressantes des espaces de voxels.

Le processus de modélisation d'une plante dans l'espace de voxels est basé sur la recherche d'une trajectoire satisfaisant un certain nombre de règles. À chaque itération, de nouvelles positions pour les extrémités de la plante sont sélectionnées selon une méthode de Monte-Carlo. C'est-à-dire qu'un certain nombre de directions de croissance sont échantillonnées, celles satisfaisant le mieux les contraintes exprimées par les règles étant alors choisies.

Les règles peuvent contraindre la croissance de façon à éviter et à rester à proximité des obstacles (un mur par exemple) ou à choisir une position d'ensoleillement maximum.

Malgré une structure ramifiée rudimentaire et des règles simples pour contrôler la croissance, le modèle de Greene est capable de générer des scènes complexes d'un réalisme surprenant. Le rendu d'un espace de voxels est ici effectué en construisant un modèle polygonal.

---

2. **voxel**: élément de volume. C'est à la 3D ce que le pixel (picture element) est à la 2D.

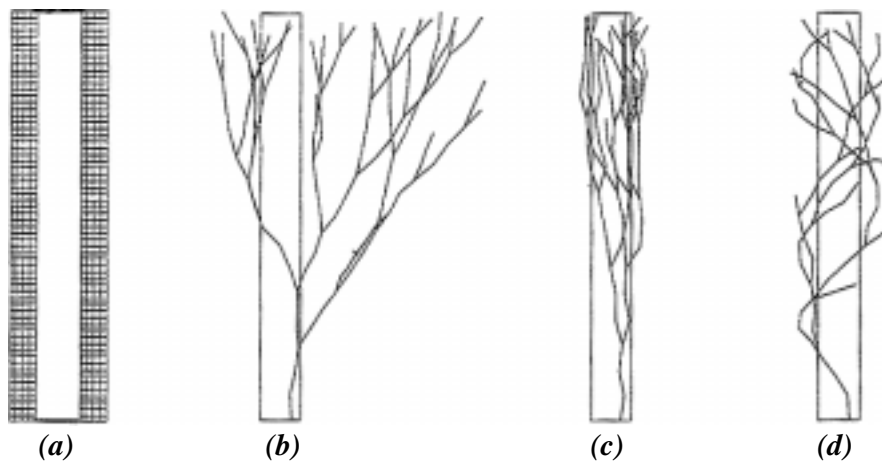


FIG. 1.25 – *Trajectoire contrainte* : (a) *l'espace de voxel est initialisé avec une forme de cylindre, chaque voxel contenant la distance au cylindre*; (b) *aucune contrainte n'est imposé à la structure*; (c) *une contrainte de proximité influence la croissance verticale*; (d) *la croissance est hélicoïdale [Gre89]*.

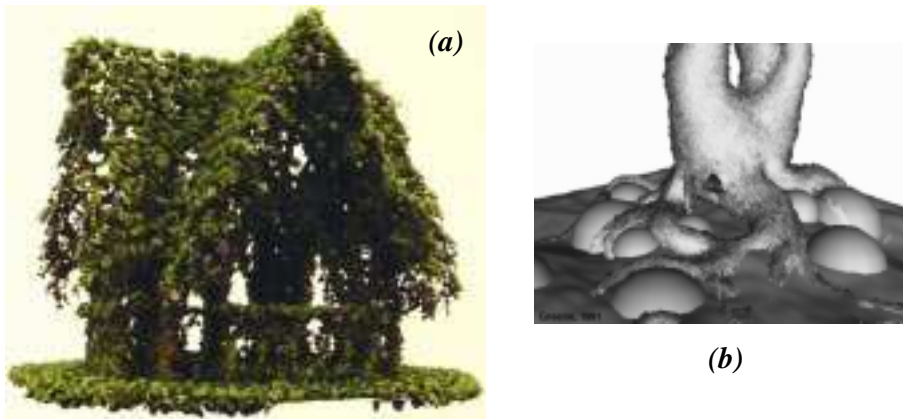


FIG. 1.26 – *Exemples de résultats* : (a) *croissance d'un lierre sur un modèle polygonal*, (b) *modélisation de racines dont la forme est contrainte par le sol et les rochers [Gre89]*.

L'utilisation d'un espace de voxels permet de prendre en compte efficacement un certain nombre de contraintes comme les collisions (entre les branches et avec l'environnement) ou la recherche de la lumière. Un espace de voxels peut également être traité rapidement par des techniques de rendu volumique efficaces. Cependant, pour une application de modélisation des végétaux, ce modèle n'est pas suffisant. Par exemple, les caractéristiques structurelles de l'architecture de la plante ne sont pas spécifiées. Ce système devrait donc être étendu par un module chargé de générer la structure ramifiée spécifique à un végétal considéré.

#### 1.1.2.4 Les modèles à base de système de particules [RB85]

Les systèmes de particules ont été utilisés pour la première fois par Reeves [Ree83, RB85] pour la modélisation et le rendu d'éléments naturels. L'idée est de composer un objet très complexe à partir de primitives très simples mais très nombreuses. Depuis plusieurs années, les particules sont de plus en plus utilisées dans le domaine de la synthèse d'images et de l'animation, en particulier dans les modèles physiques [LJF<sup>+</sup>91, DG95, LP95].

En 1983, Reeves utilise des systèmes de particules pour modéliser et animer certains phénomènes naturels comme le feu, les nuages ou l'eau. En effet, ces éléments paraissent très difficiles à représenter à l'aide des primitives géométriques classiques, d'autant plus que leurs formes et leurs caractéristiques peuvent changer au cours du temps.

Reeves propose de visualiser les trajectoires de particules pour rendre des effets de feu. Les caractéristiques et les comportements des particules sont simples, mais du fait de leur très grand nombre, des formes et des animations complexes sont créées. Les attributs de ces particules sont les suivants :

- la position dans l'espace;
- le vecteur vitesse;
- la taille de la particule;
- la couleur et la transparence;
- la forme de la particule;
- l'âge de la particule.

Une fois les valeurs initiales des paramètres déterminées, la particule évolue dans l'espace en subissant différentes influences (gravité par exemple) qui vont modifier ces attributs. Le rendu est très simple : à chaque itération, chaque particule ajoute sa contribution élémentaire aux pixels de l'image qu'elle couvre.

Dans cette première approche, les collisions entre les particules ainsi que le problème des particules cachées par d'autres sont négligés. Ces simplifications qui peuvent paraître excessives, sont en fait acceptables : les interactions locales sont peu perceptibles dans un système utilisant un très grand nombre de particules.

Avec le même principe, Sims simule des effets dynamiques complexes [Sim90] (eau, explosion). Il exploite le fait que chaque particule est indépendante pour simuler sur machine parallèle des systèmes de particules importants.

Deux ans plus tard, Reeves et Blau présentent une application des systèmes de particules pour la modélisation et le rendu d'éléments naturels [RB85]. Ainsi, pour la modélisation

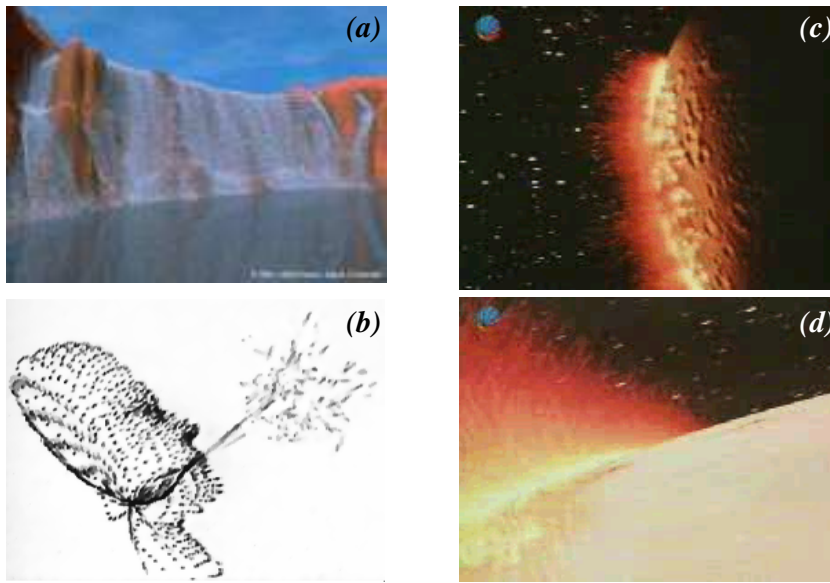


FIG. 1.27 – *Utilisation des systèmes de particules* : (a) une cascade (Miller/Gavin Apple Computer), (b) animation de particules par Sims [Sim90], (c) et (d) séquence d'explosions du film « Star Trek II: The Wrath of Khan » par Reeves [Ree83].

des arbres, ils définissent des systèmes de particules structurés dont le comportement est plus contraint que ceux utilisés dans le cas de phénomènes naturels comme le feu ou l'eau [Ree83, Sim90]. Les végétaux ont en effet une structure plus développée que des phénomènes comme le feu. Dans les systèmes de particules structurés, les particules ne sont plus strictement indépendantes mais peuvent avoir des relations entre elles.

Reeves et Blau utilisent un algorithme récursif pour générer la structure principale de l'arbre. Des paramètres géométriques comme la hauteur moyenne de l'arbre, la largeur du tronc, les angles de branchement ou la longueur moyenne des branches sont utilisés pour caractériser l'espèce de l'arbre. À chacun de ces paramètres, une valeur moyenne et un écart type sont attribués ainsi qu'une probabilité de ramification pour les branches. Quant au feuillage, les paramètres définis sont la position, la densité, l'orientation ou la couleur.

L'algorithme récursif génère donc un nombre important de particules représentées par des segments pour les branches ou des disques pour les feuilles (plusieurs centaines de milliers de particules pour un arbre). Un problème crucial se pose alors pour la mise en œuvre et le réalisme du rendu d'un tel modèle. Reeves et Blau proposent une méthode de rendu probabiliste: pour une particule, une probabilité d'ensoleillement est calculée à partir de sa position dans l'arbre, de la forme générale de l'arbre et de la position du soleil (cf. figure 1.28). Il est ainsi possible d'estimer pour chaque particule devant être rendue, une approximation de son illumination. Du fait du très grand nombre de particules composant un arbre, l'impression visuelle globale est excellente, même si localement, le rendu est approximatif.

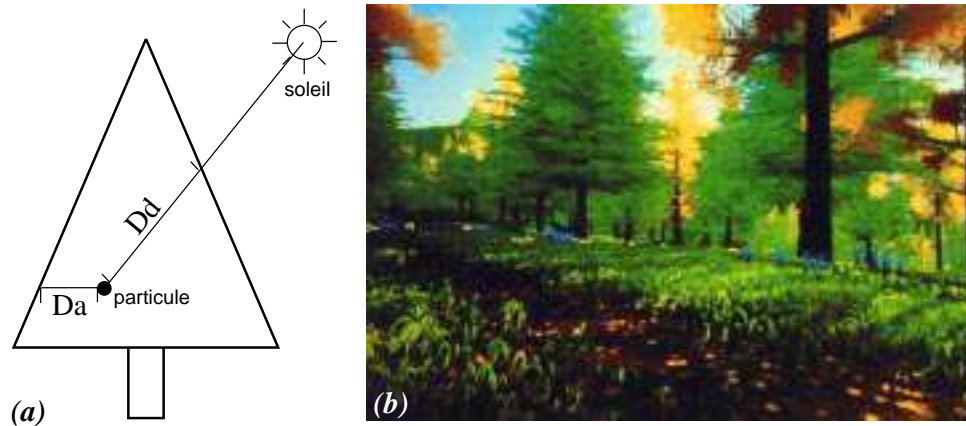


FIG. 1.28 – *Le shading probabiliste : (a) les valeurs d'illumination ambiante et diffuse pour une particule sont déduites des distances  $D_a$  et  $D_d$ . Image d'une forêt extraite du film « The Adventures of André and Wally B. » réalisé par W. Reeves et R. Blau [RB85] (voir aussi planches couleurs page 173).*

De même, le problème de la visibilité des particules est résolu par une méthode approximative. Lors de la modélisation, les particules sont grossièrement classées en fonction de leur distance à l'observateur et ensuite elles sont dessinées de l'arrière vers l'avant. Dans une scène de paysage composée de plusieurs arbres, la modélisation, le shading et le tracé sont effectués pour chaque arbre indépendamment et ceci, de l'arbre le plus lointain à l'arbre le plus proche.

Les travaux de Reeves mettent en évidence une propriété intéressante des systèmes de particules : quand le nombre de particules est élevé, des méthodes probabilistes et approximatives sont suffisantes pour résoudre les problèmes de l'illumination et du rendu.

Des résultats peuvent être vus dans le film « The Adventures of André and Wally B. » produit par Lucasfilm. Le résultat de la modélisation et du rendu des arbres est très convaincant visuellement, même si on remarque que la structure des branches sous-jacentes est simplifiée. De ce fait, une des limitations de ce modèle est son incapacité à modéliser une structure ramifiée correcte pour une espèce d'arbre donnée. Le nombre restreint des paramètres géométriques ainsi que l'algorithme de génération ne permettent pas de définir précisément une structure ramifiée spécifique.

Il est probable que les arbres de Reeves apparaîtraient bien décevants s'ils n'avaient plus de feuilles (par exemple les arbres à feuilles caduques à l'automne).

### 1.1.3 Classification et comparaison

Dans cette partie, nous tentons d'évaluer les différentes méthodes de modélisation et de rendu de végétaux. Dans un premier tableau récapitulatif, nous résumons quelques une



des caractéristiques importantes des modèles que nous avons présentés (cf. tableau 1.1). Dans un second tableau, nous donnerons une appréciation de chacun de ces modèles par rapport aux critères définis en introduction et qui nous semblent importants : le réalisme des images produites, la diversité des espèces possibles, le niveau des connaissances requises à la définition d'un objet, la complexité et le volume des calculs nécessaires et la possibilité de prendre en compte des influences externes à l'objet (cf. tableau 1.2).

Modèle	Génération de la structure ramifiée	Géométrie et rendu	Adaptation multi-échelle
Aono [AK84]	explicite, à partir de caractéristiques géométriques mesurées	segment de droite	non
Smith [Smi84]	génération par grammaires de réécritures parallèles	cylindres et sphères, méthodes de rendu classiques	non
Oppenheimer [Opp86]	structure fractale avec paramètres stochastiques	prismes polygonaux texturés, méthodes classiques	non
de Reffye [dREF <sup>+</sup> 88]	moteur de croissance botanique	primitives polygonales, Z-buffer/méthodes classiques	non
Prusinkiewicz [PLH88, PHM93, Pru93, PL90, PJM94]	moteur de croissance, grammaires L-systems	primitives polygonales, méthodes de rendu classiques	non
Viennot [VEJA89]	construction d'arbre binaire par matrices de ramification	primitives polygonales, méthodes classiques	non
Weber [WP95]	explicite, à partir de caractéristiques géométriques	modèle polygonal, Z-buffer/méthode classique	oui
Gardner [Gar84]	pas de structure ramifiée	ellipsoïdes texturés, méthode spécifique temps réel	implicite
Greene [Gre89]	structure très simple à partir de paramètres géométriques	espace de voxels, rendu volumique	possible
Reeves [Ree83, RB85]	structure simple par récursivité à partir de caractéristiques géométriques	système de particules, méthode spécifique	non

TAB. 1.1 – *Caractéristiques des techniques pour la modélisation et le rendu des végétaux*

#### 1.1.4 Conclusion

Ce tour d'horizon des méthodes permettant la modélisation des végétaux distingue deux grandes catégories de modèles :

- les approches structurelles tentent de construire une topologie la plus précise et la plus fidèle possible de la structure ramifiée. Ce sont les méthodes à base de moteur de croissance [dREF<sup>+</sup>88, PLH88] ou celles plus spécifiques à base de caractéristiques

Modèle	Réalisme des images	Diversité des espèces	Connaissances requises	Complexité et calcul	Influences externes
Aono [AK84]	faible	moyenne	faibles	faible	oui
Smith [Smi84]	faible	faible	???	faible	non
Oppenheimer [Opp86]	moyen	faible	peu	???	non
de Reffye [dREF <sup>+</sup> 88]	excellent	très grande	botaniques	moyenne	possible
Prusinkiewicz [PLH88, PHM93, Pru93, PL90, PJM94]	très bon	grande	spécifiques	moyenne	oui
Viennot [VEJA89]	moyen	faible	peu	faible	non
Weber [WP95]	très réaliste	grande	géométriques	moyen	non
Gardner [Gar84]	faible	très faible	aucune	très faible	non
Greene [Gre89]	bon	très limitée	faibles	rendu coûteux	oui
Reeves [Ree83, RB85]	bon	assez limitée	faibles	assez complexe	possible

TAB. 1.2 – *Comparaison des techniques pour la modélisation et le rendu des végétaux*

géométriques [WP95]. Ces méthodes peuvent être complexes et difficiles d'accès, mais elles génèrent des objets très proches des végétaux réels.

- les approches impressionnistes s'intéressent plus à l'aspect du végétal qu'à la qualité de la structure ramifiée. Cette structure existe toujours mais elle est en général très peu développée et limitée aux éléments perceptibles. Les modèles impressionnistes sont en général plus spécifiques et ne s'intéressent qu'à une gamme limitée de végétaux.

Quatre modèles me semblent plus intéressants de par leurs principes et leurs résultats : les grammaires L-systems, le moteur de croissance botanique, les modèles à caractère géométrique et les systèmes de particules.

Le modèle de Prusinkiewicz et Lindenmayer basé sur un système de réécriture parallèle a fait ses preuves et de nombreuses variations et extensions de ce modèle existent. Prusinkiewicz souligne d'ailleurs que quasiment tous les autres modèles peuvent s'exprimer à l'aide du formalisme des L-systems étendus.

Le moteur de croissance du logiciel AMAP [dREF<sup>+</sup>88] a lui aussi prouvé sa capacité à modéliser fidèlement une vaste gamme de végétaux. Du fait des fondements botaniques du modèle, une grande précision est possible lors de la simulation de la croissance. De nombreux paramètres ont été inclus dans le processus comme les influences externes ou les traumatismes.

L'approche de Weber [WP95] est basée entièrement sur la description des paramètres géométriques de la structure ramifiée. Des résultats très bons sont obtenus mais nécessitent la spécification d'un nombre important de paramètres.

Malgré la simplicité du modèle de Reeves et Blau [RB85], leurs images de forêt donnent

une excellente impression. L'utilisation de système de particules pour la modélisation et le rendu<sup>3</sup> semble donc une bonne approche. Le modèle développé dans ce rapport est semblable à celui de Reeves avec un certain nombre d'extensions qui permettent un plus grand contrôle sur la structure ramifiée.

---

3. la particule sert de primitive aussi bien au niveau de la modélisation que du rendu. C'est un avantage qui évite en général d'avoir une phase coûteuse d'interprétation géométrique.

## 1.2 Le relief

Comme pour les végétaux, la modélisation et la visualisation des reliefs naturels a toujours été un domaine actif de la recherche en synthèse d'images. Ce problème est d'autant plus étudié qu'il a de nombreuses applications en géographie, aménagement du territoire, exploration spatiale, simulateur de vol, etc. Comme pour certaines techniques de synthèse des végétaux, il existe deux aspects distincts de ce problème : d'une part la génération ou l'amplification des données qui définissent le terrain et d'autre part le rendu d'un tel modèle. L'aspect modélisation du terrain n'est pas toujours abordé car il est possible de disposer de données explicites : les modèles numériques de terrain issus de la mesure de reliefs réels. Dans ce bref état de l'art, nous ne nous intéresserons pas aux différentes techniques pour la mesure et le codage de données réelles, mais plus aux méthodes de génération de relief artificiel et aux techniques adaptées à leur rendu.

### 1.2.1 Revue des méthodes pour la modélisation de terrains

En 1968, Mandelbrot remarque la similitude des formes entre la crête des montagnes et la courbe produit par un mouvement Brownien fractionnaire (fractional Brownian Motion ou fBm) [Man83]. Le mouvement Brownien est essentiellement une suite de tout petits déplacements mutuellement indépendants et isotropes (toutes les directions ont la même probabilité). Le mouvement Brownien fractionnaire est la généralisation du mouvement Brownien correspondant à une dimension non entière (fractale). Ainsi, dans le plan par exemple, un mouvement Brownien fractionnaire de dimension comprise entre 1 et 2 sera plus régulier qu'un mouvement Brownien. Trois méthodes ont été proposées par Mandelbrot pour calculer une approximation discrète du fBm en plusieurs dimensions. Ce sont les champs de Poisson fractionnaires (fractional Poisson field), les approximations d'un bruit Gaussien discret (discrete fractional Gaussian noise) et la transformation de Fourier d'un bruit blanc.

Une implémentation réussie de ces techniques a été réalisée par J. Beale dans son programme *Gforge*<sup>4</sup>. *Gforge* utilise les techniques de synthèse spectrale par transformée de Fourier inverse décrites par Voss [Vos85]. Une table de valeurs aléatoires dans le domaine des fréquences est construite à l'aide d'un bruit Gaussien d'amplitude  $(1/f)^{(4-dimension)}$ . Le paramètre *dimension* représente la « rugosité » de la surface mais sans pour cela correspondre à la dimension fractale. Des filtres passe-bas et passe-haut sont ensuite appliqués à la distribution des fréquences avant de réaliser la transformée de Fourier inverse. Pour finir, les altitudes ainsi obtenues sont déformées par une fonction non-linéaire contrôlée par le paramètre *power*. La figure 1.29 présente un échantillon de surfaces obtenues pour différents paramètres *dimension* et *power*.

---

4. John P. Beale, Stanford University. *Gforge* landscape generator : <http://jumpjibe.stanford.edu:8080/beale>

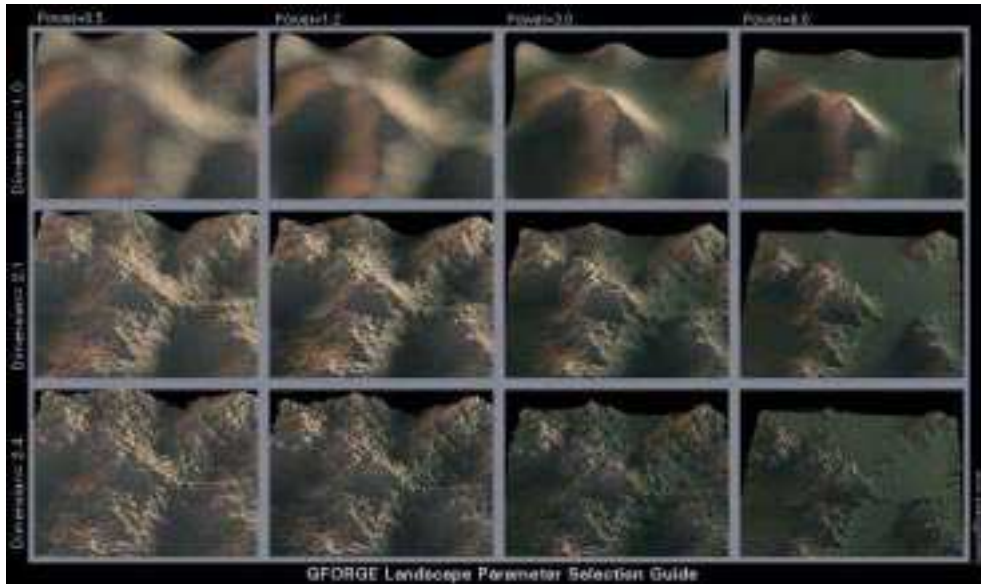


FIG. 1.29 – Exemples de reliefs produits par Gforge. Le paramètre *dimension* représente la « rugosité » et le paramètre *power* contrôle l'interprétation des altitudes.

Les méthodes à base de synthèse spectrale étant assez complexes, Fournier [FFC82] en 1982 propose une nouvelle méthode algorithmique pour le calcul approché du fBm. Cet algorithme est basé sur une subdivision récursive du modèle et sur l'introduction d'un facteur aléatoire (cf. algorithme 1.2). Le principe est le suivant : pour chaque intervalle non encore subdivisé, on calcule le point milieu auquel on applique une perturbation aléatoire dont l'amplitude est proportionnelle au niveau de récursion. Ce traitement est poursuivi récursivement sur les deux intervalles ainsi générés jusqu'à atteindre une limite fixée. La fonction stochastique utilisée pour perturber les points milieux calcule une valeur aléatoire selon une distribution Gaussienne de moyenne 0 et de variance 1.

---

**Algorithme 1.2** Algorithme de subdivision.

---

```
procedure subdivise(f1, t1, f2, t2, std : réels)
```

```
  si (t2-t1 > epsilon) faire
    tmid = (t1 + t2) / 2.0
    fmid = (f1 + f2) / 2.0 + std * gauss(tmid)
    std = std * ratio
    subdivise(f1, t1, fmid, tmid, std)
    subdivise(fmid, tmid, f2, t2, std)
  finsi
```

---

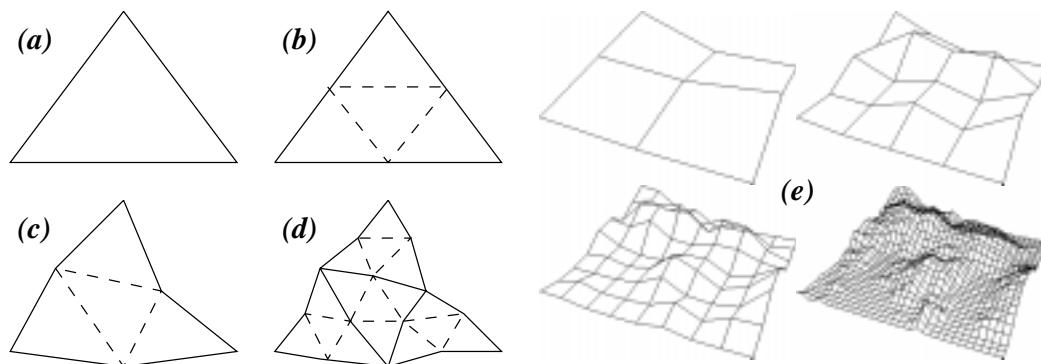


FIG. 1.30 – *Technique de subdivision*: (a) la structure 2D de base est un triangle, (c) les points milieux des côtés sont utilisés pour former 4 triangles, (c) la position de ces points est perturbée, (d) on continue ce processus jusqu'au niveau de détail requis. (e) Application de principe de subdivision sur une carte d'élévation.

Les valeurs initiales de `std` et `ratio` permettent de contrôler la dimension fractale (la « rugosité ») de la courbe produite. Mandelbrot souligne que cette méthode n'est qu'une approximation et ne possède pas les caractéristiques d'isotropisme ou d'autosimilarité des fBm. On remarque en effet un certain nombre de « défauts » sur les surfaces générées selon ce principe (cf. figure 1.32).

Pour résoudre ces problèmes et produire des reliefs plus réalistes, Miller [Mil86] définit une autre technique de subdivision. Alors que la méthode proposée par Fournier [FFC82] est « context independant » (aucune information ne circule entre les voisins), la subdivision de Miller va prendre en compte les valeurs d'un plus grand nombre d'échantillons (cf. figure 1.31) et agir ainsi comme un filtre.

Les inconvénients majeurs de cette méthode sont de deux ordres: la surface générée ne passe plus par les points de contrôle et les grilles successives sont de tailles différentes (cf. figure 1.31.b).

Les méthodes à base de subdivisions récursives sont cependant limitées et les résultats ne paraissent pas toujours très réalistes dans le cas de la génération de relief. Il apparaît en effet difficile de faire varier la « rugosité » sur une surface ou d'effectuer un contrôle local sur les caractéristiques du terrain.

Musgrave [MKM89, MM89, Mus90] remarque que dans la nature, les paysages n'ont pas une apparence régulière et uniforme: les dépressions et les reliefs âgés sont plus réguliers que les crêtes des montagnes jeunes. Afin de prendre en compte les variations hétérogènes

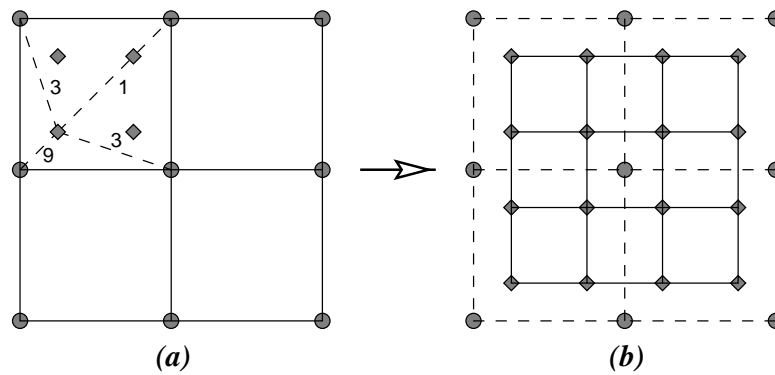


FIG. 1.31 – Principe de subdivision de Miller : (a) 4 nouveaux points sont calculés à partir de la somme pondérée des 4 points voisins du niveau précédent, (b) à chaque itération une grille de taille  $2N - 2$  est formée à partir d'une grille de taille  $N$ , Les points du bord sont perdus.

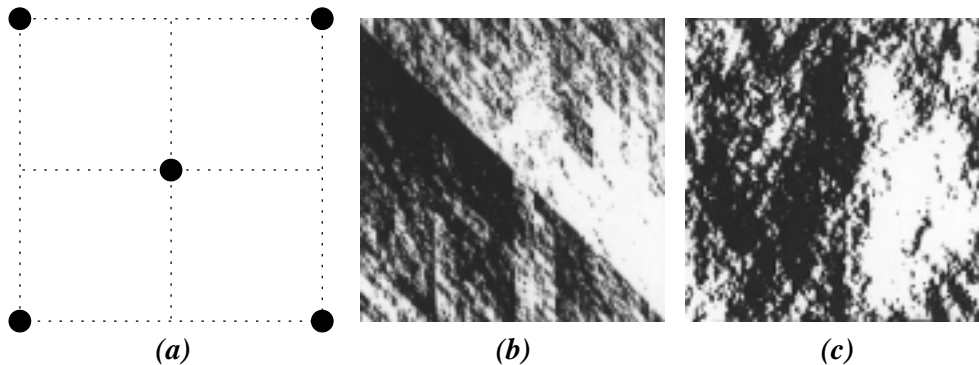


FIG. 1.32 – Résultats des méthodes de subdivision : (a) les 5 points initiaux de la subdivision correspondent aux sommets d'une pyramide, (b) résultat de la méthode de Fournier (subdivision par le point milieu), (c) résultat par la méthode de Miller (subdivision pondérée) [Mil86].

d'un même terrain, il présente une approche en deux étapes :

- génération d'un relief à partir de fonctions multi-fractales (présentant plusieurs dimensions fractales),
- application d'un processus d'érosion sur le relief.

La mise en œuvre des fonctions multi-fractales est réalisée par la somme ou le produit de fonctions mono-fractales comme les fBm. Il devient ainsi possible de sélectionner de faibles dimensions fractales pour les vallées et d'autres dimensions pour les montagnes (cf. figure 1.33).

De plus, cette méthode étant basée sur un calcul de fBm par somme de bruit, chaque

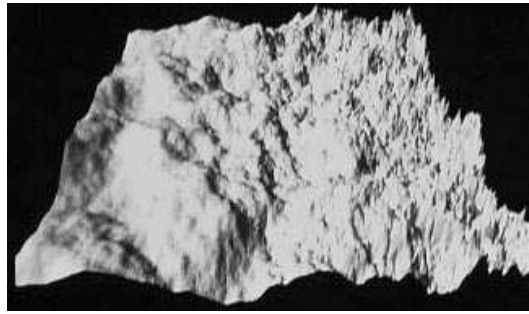


FIG. 1.33 – *Variation de la dimension fractale. La dimension fractale vaut 2.0 sur le côté gauche et 3.0 sur le côté droit du terrain [MKM89].*

point est indépendant et peut être évalué sans contrainte. Il est possible de contrôler localement la dimension fractale ainsi que le facteur d'échelle du terrain. Ce facteur d'échelle définit le rapport entre la largeur et la hauteur du relief. Par exemple, la dimension fractale des montagnes de l'Himalaya est approximativement la même que celle d'une autoroute, mais le facteur d'échelle est très différent (de l'ordre du kilomètre dans le cas de l'Himalaya et du millimètre dans le cas de l'autoroute).

Pour accentuer le réalisme des reliefs produits, Musgrave intègre un processus d'érosion dans la modélisation des terrains. Il propose un modèle simple d'érosion hydraulique: de l'eau est déposée sur chaque nœud du maillage et « ruissèle » sur le relief, érodant la matière ou déposant des sédiments sur son chemin. Il est ainsi possible de définir la fréquence et la puissance des pluies en fonction de l'altitude. Ce processus simple mais long à calculer donne d'excellents résultats (cf. figure 1.34).

Pour compléter l'effet de l'érosion hydraulique, Musgrave propose une simulation de l'érosion induite par l'action du climat. Il s'agit d'une simple technique de relaxation qui simule le transfert de matière des zones à forte pente pour former des reliefs plus réalistes.

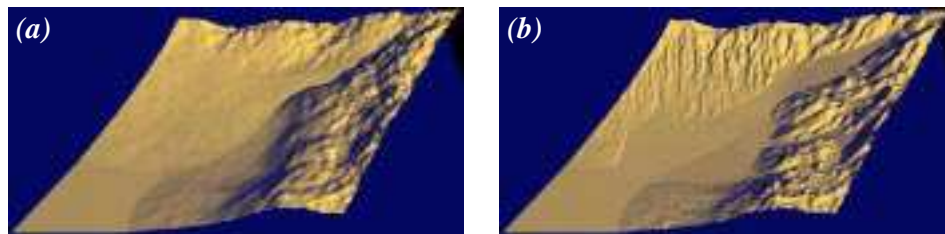


FIG. 1.34 – *Les effets de l'érosion par la méthode de Musgrave : (a) sans le processus d'érosion, (b) avec [MKM89].*

C'est également dans la catégorie des modèles par érosion que se situe les travaux de Janey [eNJ92] et de Roudier [RPP93]. Janey utilise un modèle de bassin fluvial (considéré



comme un arbre binaire planaire) pour construire par élévation un relief montagneux. Le modèle développé par Roudier est basé sur la définition d'un terrain initial avec ses structures géologiques sous-jacentes. Un procédé d'érosion pluviale permet alors de créer un terrain naturel ainsi que le réseau fluvial associé.

Ces modèles pour la création de terrains, et en particulier celui de Musgrave, sont complets et les résultats sont en général d'un réalisme surprenant. Ils peuvent cependant rester coûteux à mettre en œuvre, comme par exemple le processus d'érosion proposé par Musgrave qui peut demander jusqu'à quatre heures de calcul sur station de travail. Il reste donc à travailler, d'une part sur l'efficacité des algorithmes et d'autre part sur la prise en compte de caractéristiques précises du terrain (par exemple, les différentes matières des couches rocheuses).



FIG. 1.35 – Images d'un relief par Musgrave [MKM89] (voir aussi planches couleurs page 173).

### 1.2.2 Le rendu des reliefs

Le rendu des modèles de terrain, qu'ils soient issus de mesures photogrammétriques<sup>5</sup> ou de programmes de création de terrains artificiels, ne représente pas une grande difficulté : le relief peut être interprété comme une collection de polygones ou de surfaces paramétriques et les méthodes de rendu classique (Z-buffer ou lancer de rayon) s'appliquent alors directement. Il faut cependant remarquer que, d'une part, le nombre de primitives géométriques

---

5. **photogrammétrie** : technique de photographie stéréographique permettant de réaliser des relevés dans l'espace, en particulier les altitudes en certains points d'un relief.

nécessaires pour décrire un terrain peut être très important, et d'autre part, les représentations des modèles de terrain ont des structures géométriques spécifiques (cartes d'élévation par exemple). Il apparaît donc intéressant de développer des techniques de rendu spécifiques efficaces qui tirent avantage de ces représentations particulières.

Deux structures de données sont principalement utilisées pour représenter un terrain : les cartes d'élévation et les réseaux de polygones. Dans la suite nous présenterons rapidement ces méthodes en expliquant quelles méthodes de rendu peuvent être utilisées.

### 1.2.2.1 Les cartes d'élévation

La structure de données la plus utilisée dans le cas des terrains est la carte d'élévation. Une carte d'élévation définit, pour chaque couple  $(u, v)$  de coordonnées dans le plan, une altitude  $z(u, v)$ . Il est courant d'associer à cette carte d'élévation, une texture qui donne pour chaque nœud de la grille, la couleur  $\vec{c}(u, v)$ . Les méthodes de rendu peuvent reposer sur deux principes différents :

- échantillonnage de l'espace image : pour chaque pixel de l'image finale, la contribution des éléments de la scène est calculée. C'est le principe du lancer de rayons.
- échantillonnage de l'espace scène : on forme des facettes polygonales et pour chacune de ces primitives, on calcule la contribution sur les pixels de l'image finale. C'est le principe des méthodes de rendu direct dont le Z-buffer est une application.

Une simplification de la méthode de Z-buffer peut être envisagée dans le cas du rendu de cartes d'élévation. Il est possible de rendre les polygones sans veiller à éliminer les faces cachées par un tampon de profondeur dans le cas où les polygones sont tracés du plus loin au plus près de l'observateur. De plus, il n'est pas nécessaire de réaliser le tri des polygones (étape coûteuse), il suffit de parcourir la carte 2D des élévations selon les axes et les directions opposés au vecteur de visé (cf. figure 1.36). C'est l'application de l'algorithme du peintre au problème du rendu de carte d'élévation.

De même, le principe du lancer de rayons peut être optimisé en appliquant des algorithmes de tracé de lignes dans l'espace 2D régulier de la grille d'élévations. Le tracé de chaque rayon est réalisé par un algorithme incrémental du type DDA (analyse différentielle entière) dans l'espace des cellules de la grille d'élévation

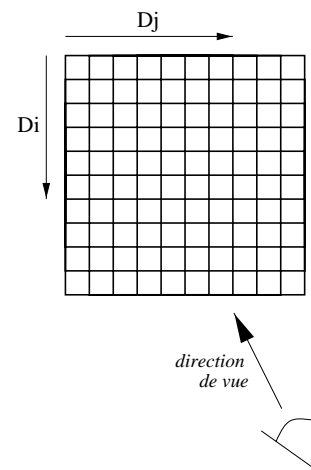


FIG. 1.36 – Le balayage des cellules s'effectue dans le sens  $D_i$  et pour chaque ligne, dans le sens  $D_j$ .

[CORLS95]. À chaque itération l'altitude courante du rayon est comparée aux altitudes des quatre points de la cellule. L'intersection peut ainsi être rapidement et facilement détectée (cf. figure 1.37).

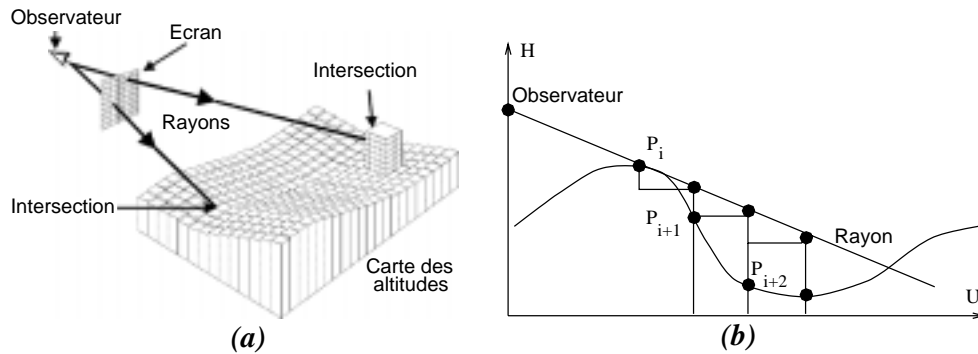


FIG. 1.37 – Lancer de rayon sur une grille d'élévation : (a) les rayons sont issus de l'observateur et parcourent la grille jusqu'à la première intersection, (b) l'algorithme incrémental trace une ligne sur la grille et détecte une intersection en comparant les altitudes [CORLS95].

Quelle que soit la technique de rendu employée, le problème de l'efficacité est posé. En effet la taille des cartes d'élévation et des textures peut être très grande dans le cas d'un vaste paysage, le volume des données à traiter est alors très important. Un certain nombre de techniques permettent de réduire ce volume en tenant compte de la position de l'observateur sur la carte et du niveau de précision demandé. Il s'agit principalement de mettre en œuvre une hiérarchie adaptative des régions, sous forme de quadtree par exemple. Ensuite, au moment du rendu, les résolutions du modèle de terrain et de la texture sont choisies en fonction de la distance à l'observateur : plus l'observateur est proche, plus le modèle doit être détaillé.

### 1.2.2.2 Les réseaux de polygones

Si les altitudes d'une carte d'élévation peuvent être trivialement représentées par un maillage de triangles, il paraît évident qu'une optimisation de ce maillage est possible et nécessaire (cf. figure 1.38). Représenter un terrain par un réseau de polygones est intéressant à plusieurs niveaux : cette structure de données est plus compacte que la grille d'élévation et des nombreux moteurs de rendu sont spécialisés dans le rendu rapide de triangles.

La construction d'un maillage triangulaire (*triangulated irregular network* ou TIN) reste un problème difficile. Il est important de trouver un compromis entre la précision du maillage (le niveau de détail sur le terrain) et le nombre de triangles.

Garland [GH95] propose une classification pour les techniques de construction de maillage :

- les grilles uniformes : c'est un échantillonnage régulier des altitudes selon deux axes;

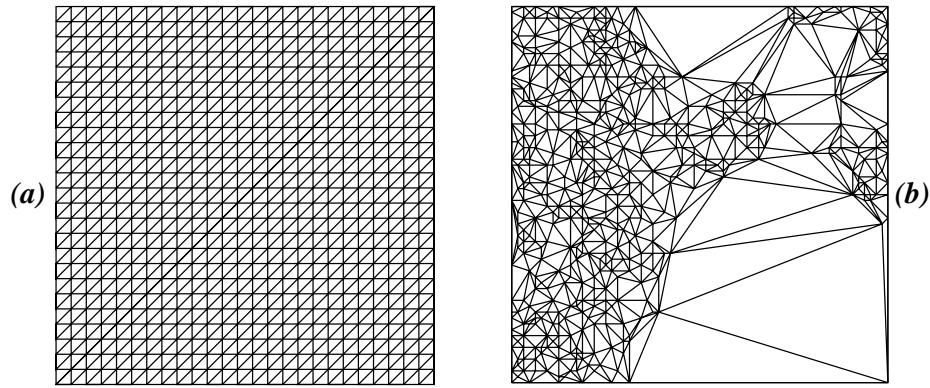


FIG. 1.38 – Un maillage de triangles (*triangulated irregular network: TIN*): (a) maillage systématique d'une grille, (b) maillage adaptatif optimisé [GH95].

- les subdivisions hiérarchiques: application des quadtrees ou des k-d trees;
- la recherche des singularités: les caractéristiques importantes du terrain (sommet, falaise, vallée ...) sont les nœuds du maillage;
- le raffinement du maillage: à partir d'une approximation minimale, de nouveaux points sont créés jusqu'au maillage final;
- la simplification du maillage: à partir du maillage exhaustif de toutes les altitudes, le réseau est simplifié;
- les autres méthodes basées sur les techniques d'optimisations.

Par extension, la construction de maillage pour des modèles de terrain rejoint le problème de la construction de bases de données pour les mondes virtuels [MGP<sup>+</sup>96]. Dans ce cas, le maillage du terrain doit être modélisé en tenant compte des objets présents sur le relief (routes, rivières, forêt et bâtiments). D'autre part, dans des environnements temps réel comme les simulateurs, il peut être efficace de générer « à la volée » un maillage du terrain. Ces méthodes à résolution adaptative intègrent les paramètres d'observation dans le processus de construction et produisent rapidement un maillage spécifique (cf. figure 1.39) [dBD95, LKR<sup>+</sup>96].

La mise en œuvre de toutes ces techniques est donc particulièrement critique dans les applications interactives. Il est nécessaire de trouver le compromis entre la rapidité des calculs (le nombre d'images rendues par seconde ne doit pas descendre en deçà d'un certain seuil) et la qualité de la restitution du relief (le plus petit détail perceptible).

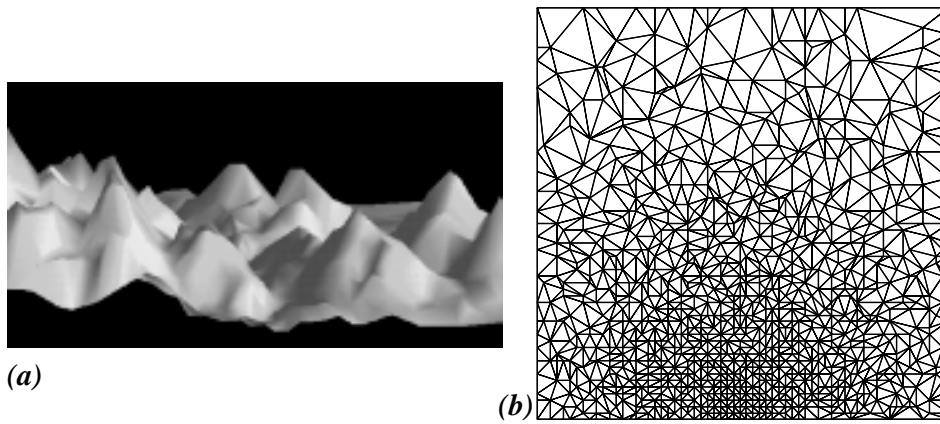


FIG. 1.39 – *Maillage adaptatif*: (a) pour un point de vue donné, (b) on génère un maillage adaptatif de triangles [dBD95].

### 1.2.3 Conclusion

Malgré les problèmes qui se posent encore, la création et le rendu de terrain sont des domaines qui ont été beaucoup étudiés et les solutions proposées aujourd’hui semblent avoir atteint un niveau convenable de souplesse et de réalisme. Des auteurs comme Musgrave étudient maintenant l’implémentation de modèles physiques pour une simulation encore plus précise de l’érosion ou pour la visualisation de l’eau. Une partie importante des recherches est également consacrée à l’optimisation de ces méthodes avec pour objectif leur utilisation dans des environnements temps réels de réalité virtuelle.

## 1.3 Les phénomènes atmosphériques

Les phénomènes atmosphériques jouent un rôle important dans le réalisme d’un paysage. Ils permettent d’intégrer dans une scène des effets importants comme la profondeur de champ, la qualité de l’atmosphère, l’heure du jour et les conditions climatiques.

Trois classes de phénomènes essentiels doivent être pris en compte dans des scènes de paysages naturels :

- **les nuages, brumes et brouillards.** Des modèles empiriques et des modèles physiques ont été développés pour la simulation visuelle de ces éléments naturels. Les plus simples utilisent les possibilités standards d’un moteur de rendu (les textures par exemple), tandis que les plus complexes modélisent la diffusion multiple de la lumière dans des milieux participatifs. Les contraintes antinomiques de réalisme et de vitesse de calcul permettent de faire le choix d’une méthode adaptée à nos besoins.
- **la couleur du ciel et la diffusion atmosphérique.** Le ciel apparaît souvent uniformément bleu dans les images de synthèse. La réalité est plus complexe : la couleur du ciel dépend de la direction du regard, de la position du soleil, des caractéristiques de

l'atmosphère et du type de terrain. Les caractéristiques de cet éclairage naturel sont également nécessaires pour un rendu réaliste de l'apparence des objets composant la scène. Des modèles complexes ont été développés pour calculer la couleur du ciel en fonction des phénomènes de diffusion de la lumière par les particules de l'atmosphère.

- **les conditions climatiques comme la pluie ou la neige.** Les phénomènes de pluie ou de neige ont, dans une scène de paysage, une influence visuelle à deux niveaux : d'une part l'effet créé par la chute des gouttes de pluie ou des flocons de neige, et d'autre part les modifications qu'ils entraînent à la surface du sol ou sur les éléments de végétation. Peu de travaux ont été entrepris sur ces deux aspects, mais des techniques existent pour modéliser et rendre des effets convaincants.

### 1.3.1 Les nuages et brumes

#### 1.3.1.1 Les modèles physiques

Un des premiers auteurs à s'intéresser à la simulation des nuages et des milieux participatifs fut Blinn en 1982 [Bli82]. Le modèle présenté simule statistiquement le passage de la lumière à travers un nuage de micro-particules. Pour simplifier le problème, Blinn ne s'intéresse qu'à une diffusion primaire de la lumière par les particules.

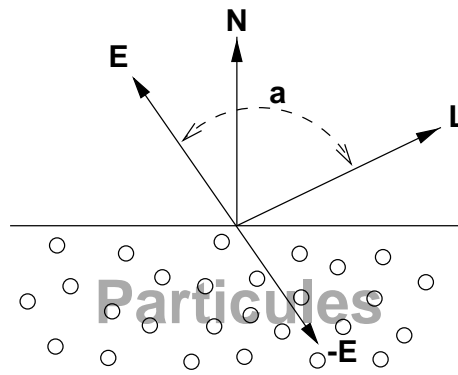


FIG. 1.40 – *Modèle de nuage.*

La quantité de lumière émise dans la direction  $E$  est calculée à partir de la diffusion par les particules de la lumière provenant de la direction  $L$ , mais aussi de la lumière provenant de la direction  $-E$  et traversant le milieu (cf. figure 1.40). Blinn formule ainsi l'intensité de la lumière émise :

$$B = (w/(N.E)\varphi(a)S) + (Tr * Bkg),$$

avec  $w$  représentant l'albédo<sup>6</sup> des particules et  $\varphi(a)$  la fonction de phase définissant la réflexion de la lumière par une particule en fonction des directions d'éclairage et d'observation. Cette fonction dépend des caractéristiques des particules que l'on veut modéliser.  $S$

6. **albédo** : fraction de la lumière reçue diffusée par une particule.

est la probabilité de diffusion liée à l'ombrage et à l'occultation des particules entre elles,  $Tr$  la transparence représentant la quantité de lumière traversant le milieu sans être arrêtée par les particules et  $Bkg$  la contribution lumineuse de la direction  $-E$ .

Le premier terme de cette formule représente la lumière diffusée et le second la lumière traversant le milieu. La fonction de phase peut être très simple pour un milieu parfaitement isotropique ( $\varphi(a) = 1$ ) ou plus compliquée dans le cas d'une diffusion de Rayleigh.

Blinn a appliqué ce modèle à la simulation visuelle des anneaux de Saturne. L'auteur remarque cependant qu'un tel modèle de diffusion primaire n'est utilisable qu'avec des particules de faible albédo (les diffusions multiples entre les particules étant négligeables dans ce cas).

En 1984, Kajiyama [KV84] développe les travaux de Blinn et propose un modèle plus général de diffusion multiple ainsi qu'une méthode de lancer de rayon permettant de rendre des objets volumiques semi-transparents. Les équations proposées sont assez complexes et le rendu d'un tel modèle peut être très coûteux en temps de calcul. Deux phases permettent de prendre en compte la diffusion multiple : pour chaque voxel du milieu on calcule la contribution et la diffusion de la lumière, ensuite on déduit la lumière incidente à un point de vue donné en parcourant cette grille 3D. Kajiyama présente également un modèle physique intéressant pour simuler la formation de nuages. Ce modèle permet de calculer sur une grille 3D la densité de particules d'eau dans l'atmosphère.

Le modèle le plus complet à ce jour semble être celui de Nishita [NMN87, NSTN93, DNKY95, NYE96]. En effet, outre la simulation de la diffusion anisotropique multiple par les particules du nuage, il prend en compte la contribution lumineuse de l'ensemble du ciel, de la lumière réfléchie par le sol et aussi de la diffusion de la lumière par les particules de l'atmosphère avant et après la traversée du nuage (cf. figure 1.41).

Cette méthode utilise une subdivision du volume en éléments homogènes (voxels) pour lesquels l'énergie reçue et émise va être calculée. Les facteurs de forme entre ces éléments de volume sont évalués en tenant compte de la fonction de phase des particules. Cette fonction de phase étant fortement directionnelle, il est possible de réaliser un certain nombre d'approximations et de pré-calculs destinés à accélérer le processus de résolution. Ainsi, Nishita résout le problème des diffusions multiples jusqu'au troisième niveau.

La forme des nuages est modélisée par des blobs (surfaces implicites) dont la surface est perturbée par une fonction fractale (cf. figure 1.42). D'excellents résultats ont été obtenus par ce modèle et les auteurs remarquent que ce principe de simulation pourrait aussi être appliqué à la visualisation de la neige.

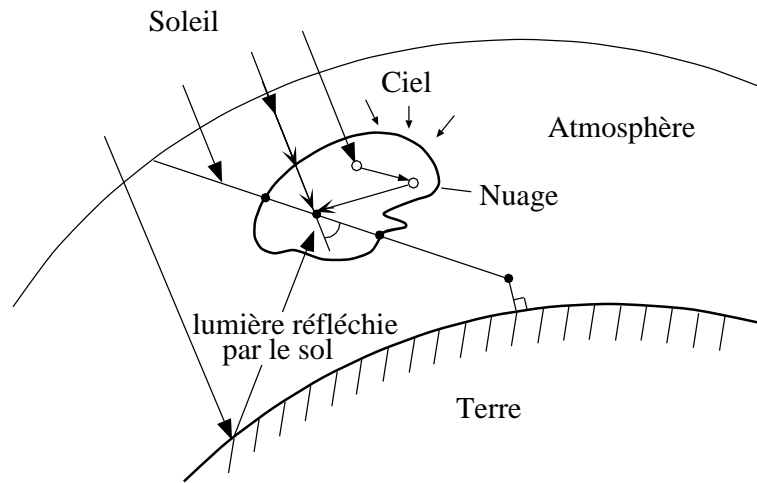


FIG. 1.41 – *Modèle de Nishita pour le rendu de nuages. Les contributions du soleil, du ciel, du sol sont prises en compte pour le calcul de la diffusion multiple.*

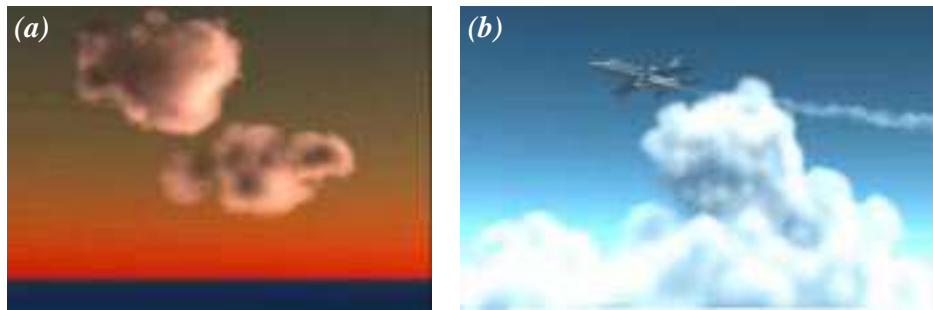


FIG. 1.42 – *Exemples de résultats : nuages sous différentes conditions d'éclairage [NYE96] (voir aussi planches couleurs page 173).*

### 1.3.1.2 Les modèles empiriques

Voss et Gardner ont proposé des modèles empiriques basés sur les textures fractales pour la visualisation de nuages [Vos85, Gar85]. Par opposition aux modèles physiques, ces méthodes permettent un rendu rapide tout en gardant un **réalisme visuel** acceptable.

Voss [Vos85] utilise une technique de synthèse spectrale pour générer une texture fractale : des valeurs aléatoires dans l'espace des fréquences sont générées en fonction de la dimension fractale voulue, ensuite une transformée de Fourier inverse est utilisée pour obtenir des amplitudes dans le domaine spatial.

C'est à partir du principe des ellipsoïdes texturés déjà utilisés pour les reliefs et les arbres [Gar84] que Gardner construit un modèle pour le rendu de nuages (cf. 1.1.2.2, page 36). La modélisation d'un nuage est réalisée par un ensemble d'ellipsoïdes fractals dont les



caractéristiques de position, de taille et de texture seront générées par une procédure. Par exemple, dans le cas des cumulus, la forme du nuage résulte de la croissance verticale de colonnes nuageuses à partir d'une carte initiale 2D (cf. figure 1.43).



FIG. 1.43 – Nuage modélisé par un ensemble d'ellipsoïdes fractals [Gar85].

La manipulation de ces ensembles d'ellipsoïdes, permet de modéliser des phénomènes dynamiques comme l'évolution des nuages ou l'animation de fumée. Ainsi Gardner utilise un modèle météorologique basé sur le mouvement de « bulles » d'air pour simuler l'évolution d'une masse nuageuse.

### 1.3.2 Le ciel

La variation de la couleur du ciel est essentiellement due à la diffusion et l'absorption de la lumière par les particules de l'atmosphère. Deux phénomènes se combinent pour donner sa luminosité et sa couleur au ciel : la diffusion de Rayleigh et la diffusion de Mie. La première est provoquée par des particules très petites (de taille inférieure à la longueur d'onde de la lumière incidente) comme les molécules de l'air, la seconde correspond à l'effet de particules plus grosses (de l'ordre de la longueur d'onde de la lumière) comme les aérosols.

Le calcul exact des caractéristiques d'un ciel est un problème difficile. En effet, il dépend de la diffusion multiple de la lumière du soleil dans les couches de l'atmosphère, mais aussi de l'éclairage indirect induit par la réflexion de la lumière sur la terre. Le phénomène de diffusion est bien sûr dépendant de la longueur d'onde de la lumière et des propriétés globales et locales de l'atmosphère. Nishita [NDKY96] propose un modèle convaincant pour le calcul de la couleur du ciel. L'équation suivante représente l'intégration des diffusions primaires

de Rayleigh et Mie pour un observateur situé en  $P_v$  et regardant en  $P_a$  (cf. figure 1.44) :

$$I_v(\lambda) = \int_{P_v}^{P_a} I_s(\lambda) R(\lambda, s, \theta) e^{(-t(s, \lambda) - t(s', \lambda))} ds$$

avec

$$R(\lambda, s, \theta) = K_r(\lambda) \rho_r(s) F_r(\theta) + K_m(\lambda) \rho_m(s) F_m(\theta)$$

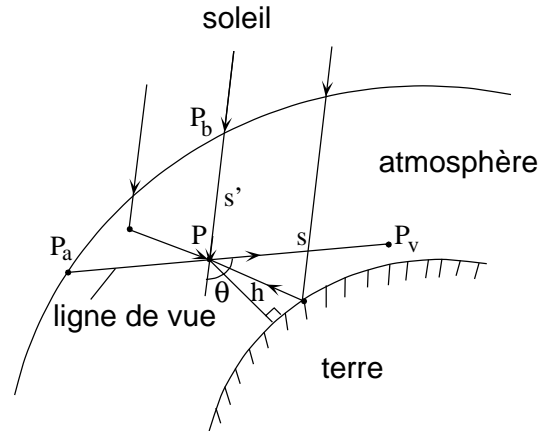


FIG. 1.44 – Intégration des contributions lumineuses pour le calcul de la couleur du ciel.

$\lambda$  représente la longueur d'onde,  $I_s$  la distribution spectrale de la lumière du soleil,  $s$  et  $s'$  les distances de  $P$  à  $P_v$  et de  $P$  à  $P_b$ ,  $K_r$  et  $K_m$  les coefficients de diffusion de Rayleigh et Mie,  $F$  et  $\rho$  les fonctions de phase et de densité associées. Les fonctions  $t(s, \lambda)$  et  $t(s', \lambda)$  réalisent l'intégration de la longueur optique de  $P$  à  $P_v$  et de  $P$  à  $P_b$ .

Pour intégrer la diffusion multiple, une méthode en deux passes est nécessaire comme dans le cas des nuages (cf. Nuages : modèles physiques, section 1.3.1.1, page 57). Dans un premier temps, la contribution directe et la diffusion multiple sont calculées sur une grille 3D de voxels par des techniques de radiativité. Ensuite, le résultat pour un point et une direction de vue donnée est calculé en collectant la lumière sur cette trajectoire (cf. figure 1.45).

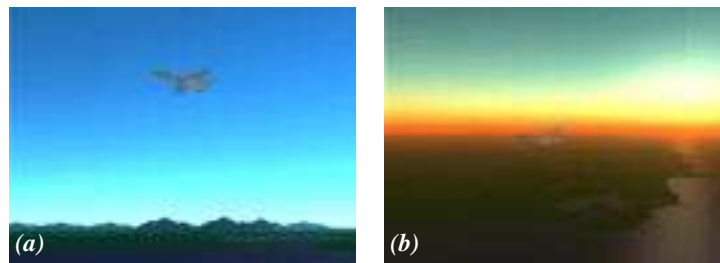


FIG. 1.45 – Exemples de ciels : (a) une vue du sol en journée, (b) une vue en altitude (4000 mètres) d'un coucher de soleil [NDKY96] (voir aussi planches couleurs page 173).

### 1.3.3 Pluie et neige

Visualiser une averse ou une chute de neige est un sujet très peu abordé en synthèse d'images. Certaines techniques comme les textures ou les systèmes de particules permettent en général une simulation satisfaisante de ces phénomènes. Ainsi, les systèmes de particules développés par Sims [Sim90] reproduisent assez bien l'impression de pluie. Par contre, une simulation plus précise et plus physique d'un tel effet constitue un problème difficile.

Des modèles pour la simulation du transport et du dépôt de la neige existent dans le domaine de la mécanique des fluides et de la nivologie. On peut citer les travaux de Masselot [MC96] qui propose une technique basée sur les particules pour simuler le déplacement et le dépôt de neige sur un terrain.

Les effets à plus long terme induits par la pluie et la neige sur le sol et les objets du paysage peuvent en général soit être traités au moment de la modélisation, soit être intégrés comme paramètres du rendu des surfaces. Ainsi la géométrie d'un terrain peut être altérée pour reproduire une couverture de neige et le rendu de cette surface devra avoir l'aspect de la neige. Nishita [NDKY96] remarque qu'un volume de neige a des caractéristiques semblables à un milieu participatif comme celui des nuages. Il propose donc des techniques basées sur l'intégration de la diffusion multiple de la lumière pour simuler une surface enneigée.

Enfin, en 1994, Reed et Wyvill [RW94] ont publié leurs travaux sur la simulation visuelle des éclairs. Le rendu de l'éclair est effectué par une méthode de lancer de rayons tandis que sa trajectoire est construite par un système de particules. Dans le cas des éclairs aussi, le rendu des objets de la scène est affecté car il doit prendre en compte cette source de lumière éphémère.



FIG. 1.46 – *Simulation visuelle d'un éclair [RW94].*

### 1.3.4 Conclusion

Ce tour d'horizon des méthodes pour la modélisation et le rendu des phénomènes atmosphériques permet de mettre en évidence deux tendances :

- l'utilisation de modèles physiques permet, quand c'est possible, un très grand réalisme des simulations. Ainsi, les modèles les plus développés doivent permettre une simulation globale et unifiée de presque tous les phénomènes atmosphériques (couleur et intensité du ciel; nuages, brumes et brouillards; neige). Cependant, dans le cas de simulations à grande échelle, les coûts de tels modèles peuvent être très élevés.
- la définition de modèles empiriques est une solution simple qui offre, en général, un réalisme visuel acceptable. Nombreuses sont les techniques qui utilisent les systèmes de particules pour produire un effet valable à moindre coût.

## Chapitre 2

# Techniques de rendu

Les techniques de rendu ont pour objectif de représenter dans l'espace image la description géométrique d'objets ou de scènes. Ce processus de rendu se divise principalement en deux phases :

- déterminer quelles surfaces sont visibles du point de vue d'un observateur
- calculer la couleur de chaque pixel de ces surfaces (éclairage des surfaces et réflexion de la lumière vers l'observateur)

Depuis les années 60, ces deux problèmes ont été très étudiés et de nombreuses solutions ont été proposées. Chronologiquement, il est possible de résumer l'évolution de ces techniques ainsi :

### Visibilité des surfaces

- algorithmes pour la détermination des lignes et des surfaces visibles travaillant dans l'espace objet (1963-1979) [Rob63].
- algorithmes à base de listes de priorités (1969-1983) [NNS72, FKN80]. Ces méthodes tentent de définir l'ordre dans lequel les objets sont dessinés. Le respect de cet ordre garantit une image correcte.
- algorithmes à balayage de lignes (1967-84) [Bou70]. Ces techniques opèrent dans l'espace image en traitant ligne après ligne les objets de la scène.
- algorithme de lancer de rayons (1968-) [App68]. Cette méthode simple calcule les objets visibles en lançant des rayons à travers tous les pixels de l'image. De par le principe de réciprocité de la lumière, cette technique simule le fonctionnement inverse d'un appareil photographique. Par extension, le lancer de rayon permet de prendre en compte de nombreuses caractéristiques optiques comme la réflexion, la réfraction, l'ombrage, etc. C'est, avec le Z-buffer, une des techniques les plus utilisées aujourd'hui en synthèse d'images.

- algorithme du Z-buffer (1974-) [Cat74]. (cf. 2.1 Le Z-buffer et ses extensions, page 65)

### **Illumination des surfaces et réflexion de la lumière**

- modèle simple d'illumination de Phong (1975) [Pho75]. Ce modèle propose une formulation empirique de la réflexion spéculaire.
- modèle de Cook et Torrance (1982) [CT82]. Cette technique est basée sur la physique et utilise les caractéristiques micro-géométriques des surfaces ou les équations de Fresnel pour déterminer la quantité et les caractéristiques de la lumière réfléchiée par une surface.
- techniques de radiosit  pour la simulation de l'illumination globale (1984-) [GTGB84]. Les techniques de radiosit  permettent de calculer les  changes d' nergie entre les surfaces composant une sc ne. La radiosit  reste aujourd'hui un domaine de recherche actif.
- mod le de Kajiya (1985) [Kaj85] pour la r flexion anisotropique.
- des mod les physiques plus sophistiqu s, par exemple celui de He [HTSG91] bas  sur l'optique physique.

Ainsi, bien que peu de m thodes diff rentes existent, de nombreuses variations et adaptations sp cifiques ont  t  propos es pour r soudre un probl me pr cis. Nous d finissons en introduction un certain nombre de contraintes impos es par la sp cificit  de notre probl me, le rendu de paysages naturels :

- la mod lisation et le rendu doivent  tre des processus simultan s. Cette contrainte impose une technique de rendu qui ne n cessite pas une description compl te. Nous excluons donc les techniques bas es sur l'espace image comme le lancer de rayon ou les m thodes   balayage de lignes.
- le rendu doit  tre r aliste, int grant l'illumination, les textures et l'ombrage tout en restant efficace. Le Z-buffer est un algorithme qui a prouv  son efficacit , en particulier par des impl mentations mat rielles. De plus, un r alisme convaincant est possible (nombreux sont les logiciels commerciaux qui utilisent cette technique) m me si un certain nombre de ph nom nes optiques sont difficilement int grables. On remarque cependant que des ph nom nes comme la r flexion sp culaire ou la r fraction sont peu pr sents dans des sc nes de paysages naturels.

Les techniques   base de Z-buffer semblent donc les plus adapt es   nos besoins. Nous d taillons dans la suite leurs principes et quelques extensions de cette m thode.

## 2.1 Le Z-buffer et ses extensions

L'origine du Z-buffer peut être attribuée à Catmull [Cat74]. Le principe initial est très simple, il s'agit en fait d'une simple extrapolation du concept de mémoire d'image. Une mémoire d'image (frame buffer) sert à stocker les attributs (la couleur) de chaque pixel de l'espace image. Le Z-buffer quant à lui est un tampon des profondeurs qui sert à stocker la profondeur de chaque pixel visible de l'espace image. En fonctionnement, la profondeur d'un nouveau pixel candidat est comparée à la valeur de la profondeur déjà stockée pour ce pixel. Si cette comparaison indique que le nouveau pixel est devant le pixel déjà stocké, ce nouveau pixel est écrit dans la mémoire d'image et le Z-buffer mis à jour (cf. algorithme 2.1).

---

**Algorithme 2.1** Algorithme du Z-buffer.

---

```

Pour chaque primitive géométrique Faire
  Pour chaque pixel de la projection de la primitive sur l'image
    Si  $z \leq Z\text{-buffer}(i,j)$  Alors
       $Z\text{-buffer}(i,j) = z$ 
      Image  $(i,j) = \text{couleur du pixel}$ 
    FinSi
  FinPour
FinPour

```

---

Comme on peut le voir, d'une part, l'ordre dans lequel sont rendues les primitives n'a pas d'incidence sur l'image finale, et d'autre part, la complexité est proportionnelle au nombre et à la surface projetée des primitives. Le Z-buffer permet donc le **rendu direct** d'une scène, c'est à dire qu'il peut traiter les objets au fur et à mesure qu'ils sont modélisés, sans stockage de description intermédiaire.

Une des implémentations les plus connues et les plus réussies de ce principe est le système de rendu REYES développé par Carpenter [Car84, CCC87]. REYES étend la qualité du rendu Z-buffer en introduisant la notion de fragment. Les contributions des primitives géométriques à la couleur d'un pixel seront définies par l'intermédiaire d'un fragment (A-buffer). Pour connaître les attributs finaux des pixels, les fragments seront composés à une résolution supérieure à celle de l'image finale. Par ce principe, il est possible d'éliminer les problèmes d'aliasing et d'intégrer des objets semi-transparents. Le système Renderman de la société Pixar qui utilise cet algorithme en a démontré les qualités.

À partir du principe du Z-buffer, de nombreuses variations et extensions ont été proposées, d'une part pour simuler certains phénomènes optiques et d'autre part pour optimiser les performances et la qualité du rendu de cette technique.

### 2.1.1 Ombres et réflexions

Le calcul des ombres portées est indispensable pour le réalisme du rendu. Dans le cas des techniques basées sur le lancer de rayons, il est facile de déterminer la visibilité des sources de lumière depuis un point quelconque de la scène. Dans le cas des méthodes à balayage de ligne ou à Z-buffer, ce problème est plus difficile. Reeves en 1987 [RSC87] a proposé une solution basée sur le calcul de cartes de profondeur depuis les sources de lumière. Ces cartes représentent les points de la scène visibles depuis une source lumineuse et donc les points éclairés. Ainsi, pour connaître l'illumination d'un point d'une surface, il suffit de consulter le pixel correspondant dans les cartes de profondeur (cf. figure 2.1).

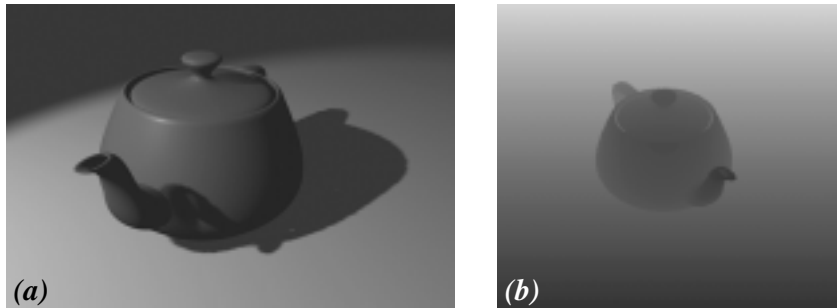


FIG. 2.1 – Calcul des ombres portées par la méthode de la carte de profondeur : (a) une image avec les ombres portées, (b) la carte de profondeur calculée depuis la source de lumière. Les points visibles sur la carte de profondeur sont les points éclairés de la scène.

L'inconvénient majeur de cette méthode est qu'elle nécessite de rendre préalablement autant d'images de la scène qu'il y a de sources lumineuses. De plus, la résolution des cartes de profondeur étant limitée, cette méthode fournit seulement une approximation des ombres portées. Des méthodes de filtrage sont indispensables pour éviter d'importants problèmes d'aliasing<sup>1</sup>.

Segal [SKv<sup>+</sup>92] a proposé une implémentation rapide de ce principe en utilisant les possibilités du matériel graphique spécialisé. Sur les stations de travail pouvant réaliser du plaquage de texture temps réel (comme les stations Reality Engine de Silicon Graphics), il est possible d'implémenter une version temps réel des cartes de profondeur pour le rendu des ombres portées.

Le rendu de miroirs ou de surfaces réfléchissantes ou semi-transparentes est également possible dans le cas du Z-buffer [Gre86b]. Comme pour le calcul des ombres portées, deux

---

1. **aliasing** : dans le cas présent, l'effet d'aliasing est dû à l'échantillonnage des objets géométriques par la grille des points de l'image lors du rendu. Il se manifeste généralement par l'apparition de crénelages aux frontières.

étapes sont nécessaires :

- **calcul d'une image à partir d'un point de vue « virtuel »**. Dans le cas d'un miroir, le point de vue sera le symétrique du point de vue de l'observateur par rapport au plan du miroir. Pour un objet à la surface réfléchissante, on calculera une image de la scène projetée sur les six faces du cube centré sur l'objet. On obtient ainsi une représentation complète de l'environnement se réfléchissant dans l'objet (cf. figure 2.2.a).
- **rendu final de la scène**. On utilise les méthodes de plaquage de texture (texture mapping) ou de plaquage d'environnement (environment mapping) pour donner l'impression d'une réflexion de la scène dans ces objets. On combine, pour chaque point de ces surfaces, les caractéristiques de l'illumination directe et les réflexions ou réfractions. La contribution de la réflexion ou réfraction est déduite des cartes d'environnements préalablement calculées (cf. figure 2.2.b).



FIG. 2.2 – Simulation de réflexion miroir avec la technique du Z-buffer: (a) 6 cartes d'environnement correspondant aux 6 faces d'un cube sont calculées, (b) les cartes d'environnement servent de texture pour la surface réfléchissante de la bouilloire [WW92].

D'autres techniques ont été proposées pour simuler avec un Z-buffer un certain nombre de phénomènes optiques. Les effets de flou de mouvement (motion blur), de profondeur de champs (depth of field) ou d'anti-aliasing ont été présentés par Haeberli [HA90]. Si ces méthodes ne fournissent pas de résultats parfaits, elles permettent cependant une bonne approximation visuelle. Dans le cadre de scènes de paysages naturels, on admet que ces techniques approchées sont suffisantes.

### 2.1.2 Optimisation du rendu par Z-buffer

Un des avantages majeurs du Z-buffer et probablement une des raisons de son succès, est la possibilité d'en réaliser des implémentations matérielles très efficaces. Aujourd'hui, les performances de tels moteurs graphiques permettent le rendu et l'affichage de plusieurs centaines de milliers de polygones en temps réel. Les applications sont très nombreuses d'autant plus que ce matériel peut être commercialisé à faible coût.



Néanmoins, pour certaines applications de synthèse d'images, le nombre de primitives graphiques à rendre est tel que des techniques d'optimisation sont indispensables. Greene [GK93] remarque que trois facteurs peuvent contribuer à accélérer le rendu :

- **la cohérence dans l'espace objet.** Il est possible de déterminer rapidement la visibilité d'un groupe de primitives voisines dans l'espace objet.
- **la cohérence dans l'espace image.** Les tests de visibilité dans l'espace image peuvent être accélérés en considérant des groupes de pixels.
- **la cohérence temporelle.** Des informations de visibilité peuvent être propagées d'une image à l'autre dans le cas d'une animation.

Si les techniques de Z-buffer exploitent naturellement la cohérence de l'espace image, elles sont moins adaptées que le lancer de rayon pour tirer parti des deux autres possibilités d'accélération. Pourtant, Greene [GK93] propose une implémentation hiérarchique du Z-buffer qui exploite les cohérences spatiales, images et temporelles d'une scène 3D. Cet algorithme utilise une subdivision spatiale en octree de la scène, une pyramide de cartes de profondeurs et une liste des nœuds visibles de l'octree. Chacune de ces structures de données permet de tirer parti des cohérences spatiales, images et temporelles.

Les primitives géométriques de la scène sont placées dans une structure d'octree comme les définit Glassner [Gla84]. Lors du rendu de la scène, si les faces d'un cube représentant une cellule de l'octree sont invisibles, alors toute la géométrie sous-jacente à cette cellule est invisible sur l'image. Le test de visibilité d'un cube est réalisé par la projection et la conversion en pixel de ces faces dans l'image courante. Si tous les pixels des faces ne sont pas en premier plan (cf. algorithme 2.1) alors, ce cube est déclaré invisible. Cette méthode permet d'éliminer du rendu des zones entières de la scène, mais elle impose la construction préalable de la structure d'octree.

Une deuxième structure de données permet d'utiliser la cohérence de l'espace image. Elle définit la carte Z-buffer comme le niveau de détail le plus fin d'une pyramide de carte de profondeurs. Chaque pixel d'une carte de profondeur de plus faible résolution va stocker la profondeur la plus importante d'un groupe de pixels. La dernière carte de la pyramide est composée d'un seul pixel qui contient la profondeur la plus élevée de l'image (cf. figure 2.3).

Grâce à cette pyramide de carte de profondeurs, il est possible de déterminer rapidement la visibilité d'un polygone dans sa totalité. Si le pixel le plus proche du polygone est plus éloigné que la profondeur du pixel qui couvre le polygone, il n'est pas visible sur l'image. Inversement, si le polygone est potentiellement visible, les nouvelles valeurs de profondeur seront propagées vers la carte de plus haute résolution pour mettre à jour la pyramide.

Pour tirer avantage de la cohérence temporelle entre deux images d'une animation, les auteurs remarquent qu'une cellule de l'octree qui était visible sur l'image précédente a une forte probabilité de l'être encore sur l'image courante. Et réciproquement, pour les cellules

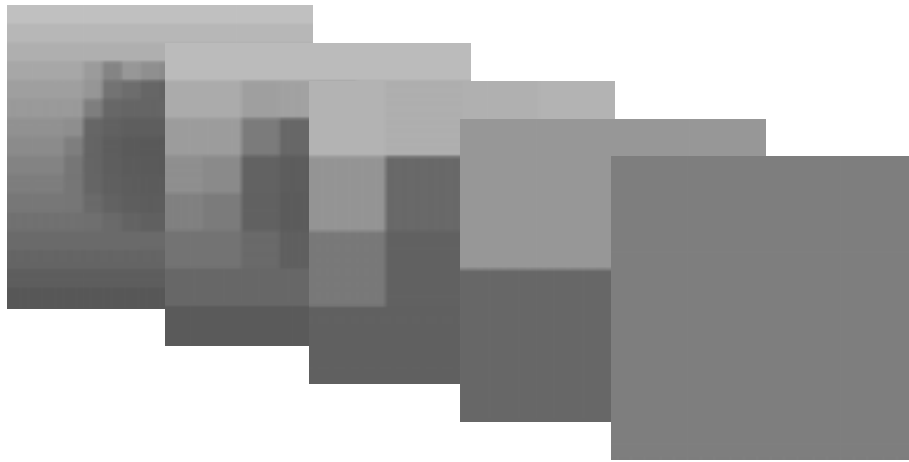


FIG. 2.3 – *Pyramide de Z-buffers : chaque carte contient la profondeur la plus importante d'un ensemble de pixels de la carte précédente.*

qui n'étaient pas visibles. Ainsi, une liste de cellules visibles de l'octree est propagée d'une image à l'autre. Ces cellules seront rendues en priorité, avant le traitement du reste de la scène. En effet, l'ordre de traitement des primitives a une importance pour l'efficacité du Z-buffer : il est préférable de rendre les objets potentiellement visibles avant les autres. Ce principe permet de minimiser le nombre de primitives qui subiront le processus coûteux de la conversion en pixels.

Les résultats présentés par cet auteurs [GK93] mettent en évidence une amélioration très significative des performances pour des scènes très importantes (jusqu'à 500 millions de polygones). On notera cependant qu'ils ne donnent aucune estimation des temps de pré-calculs nécessaires à la construction de la structure d'octree. De plus, les techniques d'optimisation basées sur des structures récursives comme les octrees nécessitent une connaissance complète de la scène. Elles ne sont donc pas compatibles avec les impératifs d'un rendu direct (sans stockage de la géométrie de la scène). Les auteurs remarquent également que de tels algorithmes ne peuvent bénéficier de l'accélération des implémentations matérielles du Z-buffer.

## 2.2 Conclusion sur les techniques de rendu

Les techniques de rendu se sont principalement développées autour de deux principes :

- **les traitements dans l'espace objet.** C'est le principe du Z-buffer, chaque objet de la scène projette sa contribution sur l'image finale.
- **les traitements dans l'espace image.** C'est le principe du lancer de rayons, on évalue pour chaque pixel de l'image finale la contribution des objets de la scène.

Ces deux techniques ont des contraintes et des avantages spécifiques. Par exemple, celles opérant dans l'espace objet permettent le rendu direct et une bonne efficacité, tandis que celles opérant dans l'espace image permettent de prendre en compte facilement une grande variété de phénomènes optiques et de mettre en œuvre des méthodes d'illumination très réalistes.

Pour répondre aux contraintes d'un système de rendu de paysages, le principe du Z-buffer a été choisi comme technique de rendu. Dans le chapitre 4.2.5 nous présenterons une implémentation spécifique de cette méthode.

## Chapitre 3

# Conclusions de l'état de l'art

Dans cet état de l'art, nous avons couvert les aspects modélisation et rendu des principaux éléments présents dans un paysage. Nous avons résumé l'ensemble des travaux portant sur la création des végétaux, présenté un tour d'horizon des modèles pour la synthèse des phénomènes atmosphériques et étudié les techniques pour la représentation, la création et le rendu de terrains. Certains travaux [PLH88, dREF<sup>+</sup>88, RB85, WP95] proposent des solutions intéressantes et convaincantes au problème de la synthèse des végétaux, mais aucun n'envisage la création de paysages naturels intégrant un très grand nombre d'objets. Notre approche de ce problème devra donc s'inspirer de ces travaux mais en restant compatible avec les objectifs énoncés dans l'introduction de ce document.

Le rendu de scène 3D étant un sujet très vaste, nous avons limité notre étude au Z-buffer et à ses extensions. En effet cette technique semble capable de répondre aux contraintes que nous avons fixées pour notre système de rendu de paysages. Nous implémenterons donc une telle méthode en tirant partie des spécificités des élément modélisés.

Cet état de l'art important nous amène à un constat : le principe de toutes les méthodes pour résoudre le problème de la modélisation et du rendu de paysages naturels existent déjà. Notre problème consiste à adapter ces méthodes pour les faire travailler efficacement ensemble.



## Troisième partie

# Les systèmes de particules pour la modélisation et le rendu



# Chapitre 1

## Introduction

L'utilisation des particules en synthèse d'image s'est beaucoup développée ces dernières années. La notion de particule a sûrement ses origines dans les sciences physiques avant d'être adaptée à d'autres domaines. Il faut cependant remarquer que le concept de particule est très vaste et doit être interprété en fonction du domaine dans lequel il est utilisé :

- en physique des particules, une particule élémentaire est un constituant fondamental de la matière ou de la lumière. Évidemment, la notion de « particule élémentaire » est en perpétuelle évolution en fonction des nouvelles théories de ce domaine.
- en mécanique, la particule désigne notamment un corps considéré comme élémentaire du point de vue du problème posé.
- en synthèse d'images, la particule est utilisée à plusieurs niveaux : depuis le concept aidant à formaliser une approche jusqu'à l'objet graphique simple et concret.

L'attrait des particules repose sur le principe suivant : des objets élémentaires aux comportements simples peuvent modéliser des objets ou des comportements complexes. En effet, si les particules sont généralement des éléments simples, les effets de leurs interactions mutuelles ou de leurs multitudes peuvent engendrer des phénomènes complexes. Par exemple, il est possible de modéliser les comportements complexes d'un écoulement à l'aide d'un nombre très important de particules élémentaires. De même, un milieu participatif peut être représenté par une multitude de particules agissant sur la lumière.

Nous définissons deux catégories d'utilisation des systèmes de particules en synthèse d'image : soit les particules ont des interactions entre elles, soit elles sont indépendantes. Dans le premier cas, les attributs et le comportement des particules dépendent des autres particules du système, c'est le principe utilisé dans la modélisation d'objets pour l'animation [LP95, LJF<sup>+</sup>91, DG95]. Dans le deuxième cas, les effets du système sont essentiellement dus au nombre très important de particules employées. Ce principe est utilisé principalement pour la simulation visuelle de phénomènes statiques ou dynamiques, comme les explosions



ou le feu [Ree83] ou les milieux participatifs [Bli82, NMN87].

Nous introduisons la définition d'un automate d'états finis pour le contrôle du système de particules. En effet, la modélisation à l'aide de système de particules indépendantes est possible si l'on est capable de spécifier des comportements précis pour les particules. C'est à ce niveau que les automates permettent un contrôle efficace et souple de l'évolution du système de particules.

Nous présentons notre approche des systèmes de particules en trois étapes. Tout d'abord, nous définirons les particules et les automates d'états finis utilisés. Nous verrons ensuite comment ces outils peuvent être utilisés pour modéliser un objet ou un phénomène. Enfin, nous présenterons l'implémentation que nous avons réalisé de ce système et l'architecture d'un moteur de rendu adapté.

## Chapitre 2

# Définitions

Deux notions sont nécessaires pour définir une particule :

- un ensemble de données définissant l'état de la particule (ses attributs);
- un modèle de comportement permettant de calculer un nouvel état pour la particule.

Le modèle de comportement est une fonction prenant en compte l'état courant de la particule, l'état du système de particules (les états des autres particules) et l'état de l'environnement dans lequel doit évoluer la particule. En résultat, cette fonction donne un nouvel état pour la particule et éventuellement un nouvel état pour l'environnement. Un modèle de comportement est donc capable de faire évoluer une particule en accord avec le système et l'environnement.

Cette définition permet d'unifier les différentes utilisations possibles des systèmes de particules. Dans le cas de systèmes de particules indépendantes, le modèle de comportement d'une particule n'est pas fonction de l'état du système mais uniquement de son état propre et de l'environnement.

Le rendu d'un tel système de particules consiste à représenter sur une image les attributs des particules.

Pour mettre en œuvre des systèmes de particules ayant des comportements complexes (comme ceux utilisés pour la modélisation des végétaux), il apparaît nécessaire de disposer d'un formalisme pour spécifier les modèles de comportement affectés aux particules, ainsi que les changements possibles de comportement. Pour ce faire, la notion d'automate d'états finis est utilisée avec les conventions suivantes :

- les états de l'automate symbolisent les comportements des particules,
- les transitions représentent les changements possibles de comportement avec éventuellement la création de particules.

Un système de particules est donc entièrement défini par les données de l'automate (topologie et fonctions associées aux états et aux transitions) et par les conditions initiales (l'ensemble des particules initiales). Le système peut ensuite évoluer librement, la création, le comportement et la mort des particules étant contrôlés par l'automate et les fonctions associées aux états et aux transitions.

Il est possible de formaliser cette approche en utilisant le principe des réseaux de transitions étendus (Augmented Transition Networks ou ATN) [Woo70, Gre86a, TG96]. Les ATN sont des extensions des automates d'états finis principalement utilisées pour le traitement des langages et la spécification d'interfaces homme-machines.

La définition formelle d'un automate d'états finis est  $M = (Q, \Sigma, \delta, q_0, f)$  avec :

- $Q$  est un ensemble fini d'états;
- $\Sigma$  est un ensemble fini de symboles;
- $\delta$  est une fonction de  $Q \times \Sigma \longrightarrow Q$  appelée fonction de transition;
- $q_0 \in Q$  est l'état initial de l'automate;
- $f$  est l'ensemble des états finaux de l'automate.

Par rapport à ce modèle, les ATN introduisent la notion d'actions associées aux états et de fonctions associées aux transitions. La définition d'un ATN est donc  $M = (Q, \Sigma, A, \delta, \gamma, F, \beta, q_0, f)$  avec en plus par rapport aux définitions précédentes :

- $A$  est un ensemble fini d'actions;
- $\gamma$  est une fonction de  $Q \longrightarrow A$  qui associe états et actions;
- $F$  est un ensemble fini de fonctions;
- $\beta$  est une fonction de  $\delta \longrightarrow F$  qui associe transitions et fonctions.

Dans ce modèle, une transition est réalisée si la fonction associée rend une valeur **VRAI** après exécution. Le principe des ATN est suffisamment riche et général pour permettre de définir des comportements spécifiques et complexes (cf. figure 2.2). Les actions peuvent être simplement l'intégration d'un mouvement particulier, l'évolution des attributs ou la destruction de la particule. Les fonctions associées aux transitions peuvent par exemple réagir

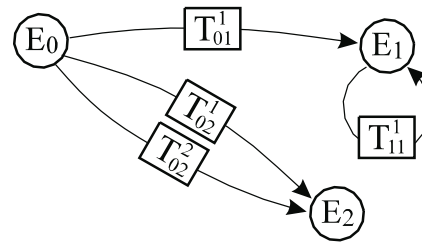


FIG. 2.1 – Un automate caractérisant un système de particules : les comportements sont représentés par  $E_i$ , les fonctions de transition par  $T_{ij}^k$ .

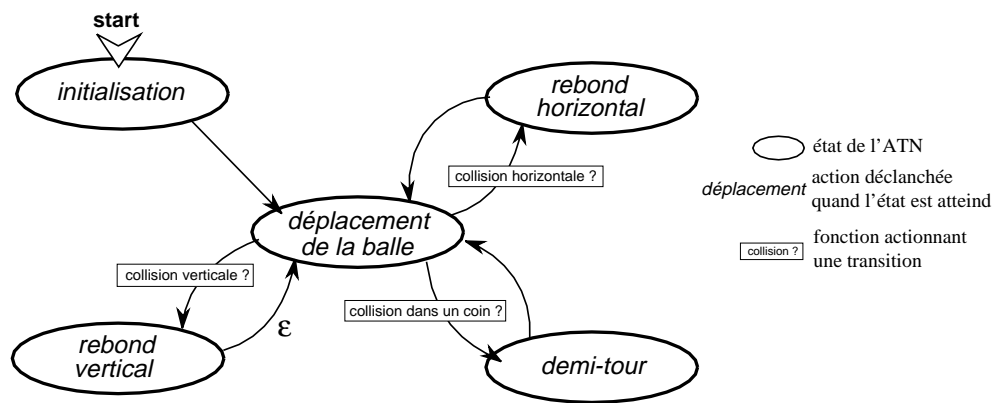


FIG. 2.2 – Exemple d'ATN: le comportement d'une balle rebondissante [TG96].

à l'environnement ou sélectionner certaines particules.

Dans le cas des systèmes de particules, l'évolution est considérée comme terminée si une action terminale est déclenchée ou s'il n'existe plus aucune particule.



## Chapitre 3

# Utilisation

Il est possible d'utiliser le couple « système de particules - automate d'états finis » à plusieurs niveaux :

- l'automate contrôle l'évolution d'une particule en spécifiant les différents comportements et transitions possibles. L'exemple de la balle rebondissante de la figure 2.2 illustre ce principe. La figure 3.1 présente une autre application de ce mode de fonctionnement.
- les transitions de l'automate peuvent créer de nouvelles particules. Dans ce cas, le nombre total de particules changera en fonction de l'évolution du système. La figure 3.2 illustre ce principe.
- l'environnement dans lequel évoluent les particules peut être modifié par les actions de l'automate ou directement par les particules.
- la topologie et les caractéristiques de l'automate sont auto-modifiées par le système.

Modéliser et rendre un objet à l'aide de cette technique consiste initialement à répondre aux questions suivantes :

- Que représente les particules par rapport à l'objet ou au phénomène? Mais aussi, que doit représenter le comportement des particules?
- Comment décrire la structure de l'objet par l'automate d'états finis? Cela revient à spécifier trois points: le rapport entre la topologie de l'objet et la topologie de l'automate, le rôle des actions associées aux états et le rôle des fonctions associées aux transitions.
- Comment rendre les particules pour obtenir une image convainquante et réaliste de l'objet?

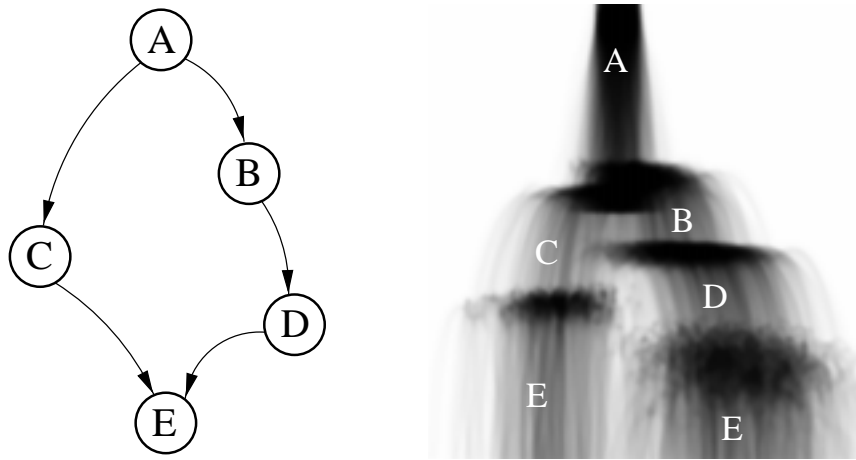


FIG. 3.1 – Les états de l'automate déplacent les particules tandis que les transitions changent les directions quand la position de la particule atteint un seuil donné.

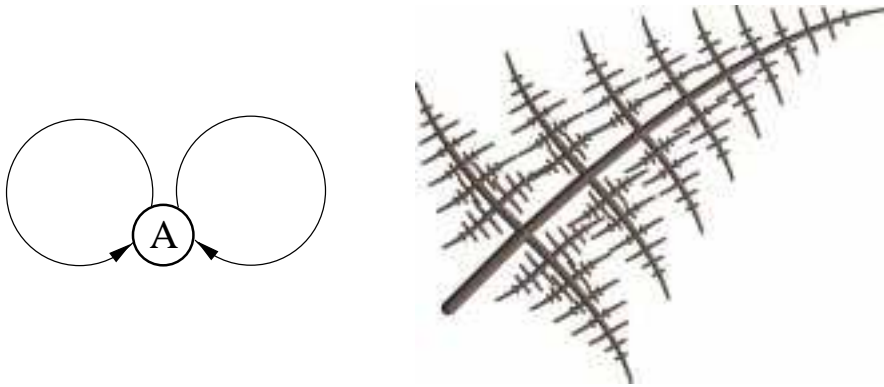


FIG. 3.2 – Structure ramifiée : les transitions créent deux nouvelles particules dans le même état. Une structure auto-similaire est générée.

Il existe toujours plusieurs façons d'aborder la modélisation d'un élément et dans notre cas, cela peut se résumer à deux approches. La première possibilité est l'analyse de la topologie de l'élément pour en déduire une structure représentative. Ainsi, la modélisation par le système de particules devra s'efforcer de reproduire le plus fidèlement possible cette structure. C'est l'approche qui a été choisie pour la modélisation des végétaux (cf. partie 2, page 103). La seconde possibilité est basée sur l'observation de l'apparence de l'élément. Dans ce cas, le système devra reproduire cette apparence en utilisant des structures et des comportements qui n'auront pas forcément de rapport avec la topologie de l'élément. La technique utilisée pour la création de nuages illustre bien cette approche (cf. chapitre 3, page 121).

---

Le rendu de ces systèmes de particules est une représentation des attributs des particules. En fonction de l'objet modélisé ou du phénomène simulé, les contributions des particules à l'image seront intégrées spécifiquement. Par exemple, dans le cas de la modélisation géométrique d'un arbre, les positions successives des particules décriront la structure et la forme des ramifications : l'intégration est spatiale. Pour la simulation d'une explosion, l'intégration des attributs des particules est temporelle.

Quelle que soit l'approche utilisée pour la modélisation des éléments, l'objectif reste toujours le réalisme visuel des images produites. Ainsi, la création d'une structure ramifiée exacte pour une espèce d'arbre n'est intéressante que si elle permet de générer un modèle ayant un impact visuel satisfaisant.





## Chapitre 4

# Mise en œuvre

L'implémentation des systèmes de particules contrôlées par automates est un aspect important de mon travail. En effet, elle doit satisfaire plusieurs critères importants :

- **efficacité** : le calcul du comportement des particules et leur rendu doivent être très rapides;
- **modularité** : il doit être possible de modifier et d'étendre facilement les fonctionnalités;
- **ressource** : pour limiter les ressources nécessaires, des méthodes d'allocation dynamique doivent être utilisées.

Le cœur de l'implémentation est décomposé en deux parties distinctes et indépendantes : construction et évolution du système de particules et rendu des particules. La première partie propose les structures permettant la définition des particules et des automates, ainsi que l'algorithme de base implémentant le moteur d'évolution. La seconde partie est un moteur de rendu dont l'architecture est prévue pour satisfaire les contraintes des systèmes de particules.

### 4.1 Implémentation du système de particules

Trois éléments principaux définissent la mise en œuvre des systèmes de particules :

- la structure contenant les attributs d'une particule;
- la structure contenant la définition de l'automate;
- le moteur d'évolution du système.

### 4.1.1 La structure Particule

La structure PARTICULE contient d'une part **les attributs standards** qui sont indispensables au fonctionnement du système et d'autre part **les attributs spécifiques** correspondant au phénomène modélisé.

#### Attributs standards

- **identification** : identifie la particule par une clef unique.
- **état courant** : définit l'état de l'automate auquel est lié la particule. Cet état définit le comportement courant de la particule.
- **particule suivante** : pointeur de liste chaînée vers la particule suivante.

#### Exemples d'attributs spécifiques

- **position courante** : donne la position courante de la particule dans l'espace 3D;
- **vitesse courante** : vecteur vitesse;
- **taille courante** ;
- **âge de la particule**.

Les attributs standards sont obligatoires tandis que les autres dépendent uniquement de l'utilisation du système de particules.

### 4.1.2 Les structures État et Transition

Deux entités sont nécessaires pour définir un automate d'états finis : une structure ÉTAT qui donne les paramètres relatifs à un état et une structure TRANSITION pour les liaisons entre états.

#### Paramètres standards d'État

- **identification** : identifie l'état par un numéro unique.
- **action** : procédure définissant l'évolution des particules dans cet état. En plus de la modification des attributs d'une particule, la procédure peut également supprimer la particule ou créer de nouvelles particules.
- **liste des transitions associées** : donne la liste des structures TRANSITION dont cet état est l'origine.

#### Paramètres standards de Transition

- **identification** : identifie la transition par un numéro unique.

- **fonction** : fonction associée à la transition et conditionnant son application. Comme dans le cas des actions associées aux états, cette fonction peut modifier les attributs de la particule.
- **état connecté** : définit l'état destination de la transition.

Des paramètres spécifiques peuvent être ajoutés pour décrire un comportement particulier, comme par exemple, des valeurs de probabilité pour les transitions.

### 4.1.3 Algorithme principal

Cet algorithme doit faire évoluer le système de particules à partir de la définition de l'automate d'états finis. Ainsi, à chaque itération, toutes les particules du système seront évaluées par l'automate (cf. algorithme 4.1).

---

**Algorithme 4.1** Algorithme principal pour l'évolution du système.

---

Pour toutes les particules du système :

Rendre la particule

En fonction de l'état associé à la particule :

Appliquer l'action associée à l'état

Activer les transitions associées à l'état :

Appliquer la fonction associée à la transition sur la particule

Si le résultat de la fonction est VRAI

Changer l'état courant de la particule

---

## 4.2 Rendu de particules

Les contraintes pour le moteur de rendu, tel que nous les avons définies en introduction de ce document, sont les suivantes :

- la modélisation et le rendu doivent être des processus simultanés. Ainsi, il n'y aura pas de description géométrique à stocker, pas de représentation intermédiaire entre la phase de modélisation et celle de rendu.
- le rendu doit être réaliste, intégrant l'illumination, les textures et l'ombrage tout en restant efficace.
- le processus complet de modélisation/rendu d'un objet doit être rapide, afin d'intégrer en un temps raisonnable, un nombre important d'éléments naturels dans une scène de paysage.

### 4.2.1 Élimination des parties cachées

Comme nous l'avons expliqué dans l'état de l'art sur les techniques de rendu (cf. 2, page 63), les techniques à base de Z-buffer sont les plus adaptées à nos objectifs. Le Z-buffer que nous avons implémenté dans le cadre de ce système possède les caractéristiques suivantes :

- la mémoire d'image stocke quatre canaux, trois sont nécessaires pour le codage de la couleur (rouge, vert et bleu) et le quatrième mémorise l'opacité du pixel (canal  $\alpha$ ). La prise en compte de l'opacité au niveau des pixels permet de résoudre les problèmes d'aliasing et partiellement les problèmes des objets semi-transparents.
- le tampon des profondeurs donne la profondeur de chaque pixel de la mémoire d'image.

### 4.2.2 Calcul de la couleur

Un autre point important de l'implémentation du moteur est le calcul de la couleur réfléchiée par les objets en fonction des caractéristiques d'illumination, de texture et de perception. Cette opération doit être très efficace car elle sera souvent sollicitée (pour le rendu de tous les pixels). Néanmoins, elle doit permettre un rendu précis intégrant les ombres, les textures et les effets de profondeur de champs. Enfin, une architecture modulaire doit permettre de modifier ou d'étendre facilement les fonctionnalités de cette partie du moteur de rendu. L'architecture retenue qui satisfait ces contraintes est inspirée du système RenderMan [CCC87, Ups89, HL90, AM90, SPS94] (cf. figure 4.1).

Les modules représentés sur la figure 4.1 correspondent aux différents types de « *shader* » que propose le système RenderMan [Ups89, HL90]. Ils forment la chaîne chargée d'évaluer la couleur finale d'un pixel d'une surface en fonction des caractéristiques globales et locales de l'environnement. Leurs fonctionnalités sont les suivantes :

- **source de lumière** : ce module doit calculer la quantité de lumière émise par une source dans une direction donnée. Il doit donc prendre en compte les caractéristiques de cette source (position et distribution), la position du point éclairé et dans le cas du Z-buffer, la carte d'ombre correspondante (cf. 2.1, Le Z-buffer et ses extensions, page 65). Dans notre application de synthèse de paysages naturels, la seule source de lumière directe étant le soleil, ce module est assez simple.
- **atmosphère 1 et 2** : les caractéristiques de la lumière peuvent être affectées lors du trajet source-objet et objet-observateur. Ce module calcule l'influence du milieu traversé sur la lumière. Ce peut être simplement une atténuation en fonction de la distance (effet de profondeur de champ) ou l'intégration de la diffusion de la lumière par des milieux participatifs comme les brouillards ou les nuages (cf. 1.3.1 Les Nuages et brumes, page 57).
- **surface** : ce module doit déterminer les caractéristiques de la surface en un point donné. Il s'agit généralement de prendre en compte la texture de la surface pour dé-

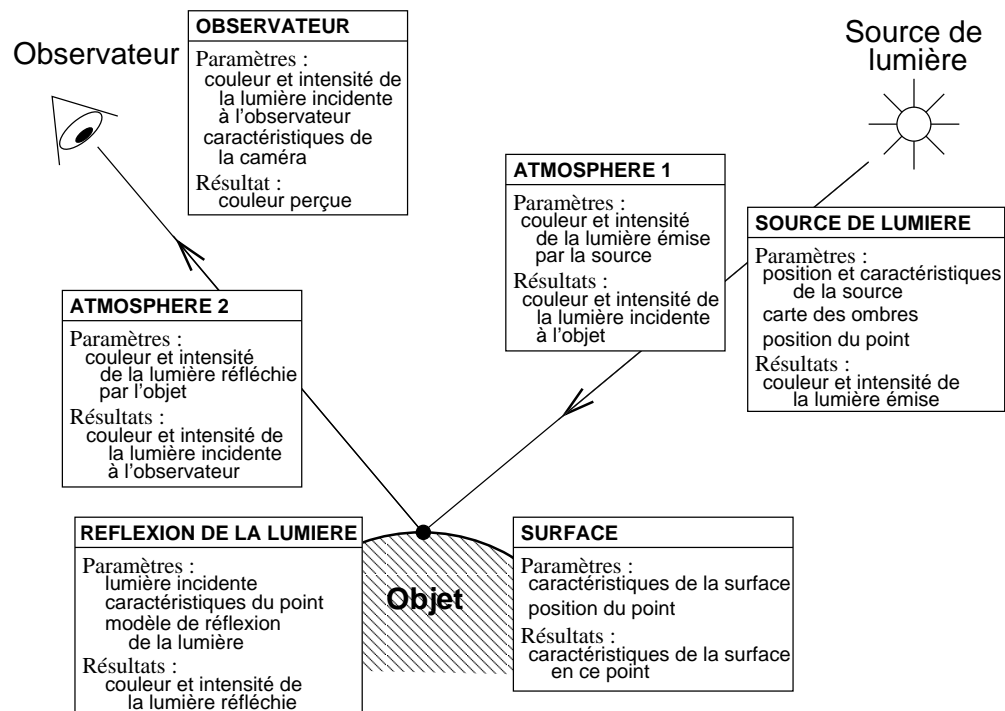


FIG. 4.1 – Les différents modules pour le calcul de la couleur d'un point visible d'un objet. Chaque module utilise des paramètres fixés pour la scène (source de lumière, observateur ...) et des paramètres dépendants du contexte (caractéristiques de l'objet, position du point ...).

terminer localement la couleur et les paramètres de réflexion (coefficients de réflexion, normale à la surface).

- **réflexion de la lumière** : ce module détermine la lumière réfléchiée dans la direction de l'observateur en fonction de la lumière incidente et des caractéristiques de la surface. Un modèle de réflexion de la lumière doit être choisi. Ce peut être simplement un modèle Lambertien pour la composante diffuse de la réflexion ou un modèle plus sophistiqué basé sur la physique (cf. 2 Techniques de rendu, page 63). Dans notre application, les réflexions de la lumière étant principalement diffuses, un modèle simple de Phong a été choisi.
- **observateur** : ce dernier module de la chaîne de rendu permet de calculer la couleur réellement perçue en fonction des caractéristiques de la caméra. Il s'agit en général de techniques de filtrage ou de conversion de couleurs.

Cette organisation en modules est pratique et puissante : elle facilite l'extension et permet de prendre en compte tous les aspects nécessaires de l'illumination des surfaces.

### 4.2.3 La chaîne de rendu

Le processus complet de rendu d'une primitive graphique est donc le suivant :

---

**Algorithme 4.2** Algorithme de rendu d'une primitive graphique.

---

```

Test de visibilité dans l'espace de la scène
Projection de la primitive dans l'espace de l'observateur (repère caméra)
Clipping de la primitive projetée par les limites de l'image
Balayage de la surface pour conversion en pixels (<< rasterization >>)
Pour chaque pixel :
    Tester la visibilité avec le Z-buffer
    Si le pixel est visible :
        Évaluer la source de lumière (Module source de lumière)
        Évaluer l'atténuation atmosphérique (Module atmosphère 1)
        Évaluer les caractéristiques de la surface (Module surface)
        Évaluer la réflexion par la surface (Module réflexion)
        Évaluer l'atténuation atmosphérique (Module atmosphère 2)
        Évaluer la couleur du pixel (Module observateur)

        Écrire la couleur du pixel dans la mémoire d'image
        Écrire la profondeur du pixel dans le Z-buffer
    Finsi
FinPour

```

---

La partie de ce processus la plus coûteuse en calcul est la conversion en pixels de la primitive géométrique. Elle doit non seulement calculer les pixels de l'image qui sont couverts ou partiellement couverts par la primitive, mais aussi, déterminer pour chacun de ces pixels la profondeur et la normale. Dans la majorité des cas, les traitements effectués par les modules sont simples et ne nécessitent pas de calculs coûteux.

### 4.2.4 Les possibilités de parallélisation des traitements

Comme dans toutes les applications très consommatrices de calculs, il est intéressant d'évaluer les possibilités de parallélisation et de distribution des traitements. Plusieurs niveaux de parallélisation sont possibles dans l'architecture que nous avons définie ci-dessus :

- traitement de l'illumination: les différents modules pour le calcul de la couleur d'un pixel étant indépendants les uns des autres, il est possible de distribuer ces calculs. Les modules ainsi distribués sur des unités de traitements indépendantes seront activés par les données qui les traversent (principe d'exécution flux de données). Un module calculera donc en parallèle avec les autres une partie du processus d'illumination (cf. figure 4.2.a) pour des pixels différents.

- traitement du rendu des primitives: le processus complet de rendu peut être divisé en plusieurs étapes indépendantes (cf. figure 4.2.a). Chacune de ces parties peut être distribuée comme le sont les modules pour le calcul de l'illumination. Une décomposition similaire du processus est réalisée dans le moteur graphique « Reality Engine » [Ake93].
- traitement parallèle par plusieurs processus de rendu: les primitives géométriques peuvent être distribuées sur plusieurs unités de traitement en fonction de leur position sur l'image. La technique du Z-buffer permet en effet de diviser l'image en plusieurs parties qui seront traitées séparément. Un processus de contrôle est chargé de distribuer les primitives à l'unité correspondante (cf. figure 4.2.b).

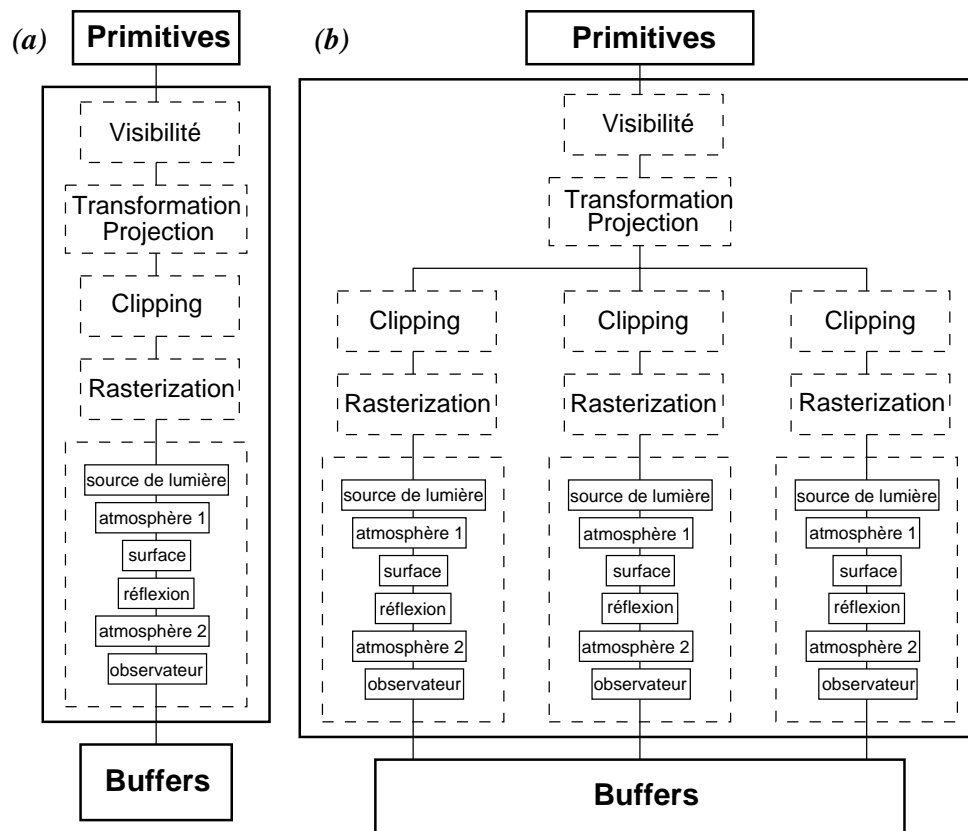


FIG. 4.2 – L'architecture modulaire du processus de rendu: (a) les modules du processus de rendu, (b) une parallélisation possible dans le cas du Z-buffer.

Ces trois configurations de distribution des calculs correspondent à trois niveaux de « granularité » croissante. Dans le cas de l'illumination, les modules sont successivement activées par les données relatives à un pixel. Dans le cas des processus de rendu distribués ce sont des primitives ou des groupes de primitives qui constituent les données de base. La granularité d'un processus distribué conditionne directement le volume des communications



entre les modules : plus les données sont élémentaires plus le volume de communication sera important. Inversement, les communications seront plus limitées dans le cas de données de plus haut niveau comme les primitives géométriques.

L'intérêt d'une technique de distribution doit être évalué en comparant la granularité de la distribution avec le support matériel disponible pour les traitements. On distingue essentiellement trois niveaux matériels de parallélisation : la parallélisation dans les composants VLSI, le multi-processeur et le réseau d'ordinateurs. Le débit des communications entre les modules d'un composant VLSI spécialisé peut être très élevé tandis qu'il est beaucoup plus faible dans le cas d'un réseau d'ordinateurs.

A titre d'exemple, l'architecture spécialisée du moteur graphique « Reality Engine » dispose de plusieurs unités de traitement pour la seule fonction des transformations géométriques (jusqu'à 12 « Geometry Engines »), de plusieurs unités de « rasterization » (jusqu'à 20 « Fragment Generators ») et d'unités de composition de pixel (jusqu'à 320 « Image Engines »). Cette architecture est efficace car elle permet des débits de communications entre les modules en rapport avec la granularité des données. Ainsi, nous pensons qu'il n'est pas rentable de distribuer les modules d'illumination ou du rendu au niveau matériel du réseau d'ordinateurs.

Les contraintes matérielles que nous avons ne permettant que la distribution des calculs sur un réseau d'ordinateurs, il n'est pas intéressant d'utiliser l'une des techniques mentionnées précédemment. La granularité des données que nous distribueront sur les différents ordinateurs sera donc encore plus grande. Comme nous le verrons dans la partie VI (page 153), notre choix est de distribuer directement la modélisation et le rendu d'éléments naturels. Ce choix est cohérent par rapport au débit de communication possible entre deux ordinateurs distants d'un réseau.

#### 4.2.5 Spécialisation du Z-buffer dans le cas de particules sphériques

Dans le cadre du rendu de systèmes de particules, la primitive géométrique la plus utilisée est la sphère (cf. section 2.2, page 104). Il est donc important que le module de « rasterization » de cette primitive soit très efficace. D'autre part, l'utilisation d'une mémoire d'image avec un canal d'opacité, ne résout pas complètement le problème des objets semi-transparents (ces objets doivent être soumis au processus de rendu dans l'ordre de leur distance à l'observateur pour que la composition de leur opacité fournisse un résultat correct).

Pour ces raisons, nous proposons une nouvelle extension du Z-buffer appelée Particule buffer (P-buffer). Le principe du P-buffer est semblable à celui du A-buffer proposé par Carpenter [Car84]. Pour chaque pixel, une liste chaînée de fragments est maintenue. Chaque fragment représente la contribution d'une particule sur ce pixel.

Pour connaître, la couleur effective du pixel, une opération de composition / compression est nécessaire : parcourir la liste chaînée, en partant des fragments les plus proches jusqu'aux fragments les plus lointains de l'observateur. Cette technique permet de résoudre un certain

nombre de problèmes normalement inhérents aux techniques de Z-buffer :

- le calcul correct de l’anti-aliasing : les pixels intègrent exactement la contribution d’une particule par rapport à la surface qu’elle couvre.
- le calcul correct de la transparence : les particules semi-transparentes peuvent être intégrées sans contraintes d’ordre dans le rendu. Il devient possible de modéliser des objets ou des phénomènes par composition et accumulation de particules semi-transparentes (cf. Modélisation et rendu de nuages, chapitre 3, page 121).

Carpenter [Car84] propose, dans la mise en œuvre du A-buffer, d’associer à chaque pixel une liste chaînée de fragments. Ces fragments définissent, à l’aide d’un masque, la couverture du pixel par la primitive graphique ainsi que les attributs de profondeur, de couleur et d’opacité. Dans le cas du A-buffer, le masque est représenté par une matrice de 4 lignes et 8 colonnes (32 points qui représentent donc un sur-échantillonnage du pixel).

Par rapport aux travaux de Carpenter, les fragments associés aux pixels du P-buffer définissent les attributs géométriques et les attributs de couleur d’une particule (dans notre cas, une sphère). Un masque de sur-échantillonnage du pixel n’est donc plus nécessaire, la couverture exacte du pixel par la particule pouvant être calculée à partir des caractéristiques de la sphère (centre et rayon) et des caractéristiques du pixel (centre et taille). Ce calcul de la surface couvrante est uniquement effectué au moment du processus de composition/compression des fragments. Le P-buffer est donc une spécialisation du A-buffer dans le cas d’une primitive unique : la sphère (cf. figure 4.3).

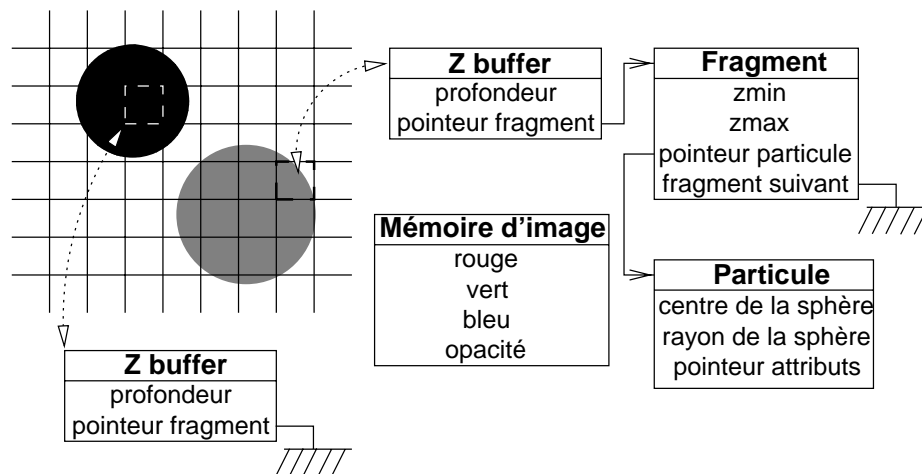


FIG. 4.3 – Les structures du P-buffer : si une particule est semi-transparente, une structure **Fragment** est créée. Ces structures définissent la profondeur du fragment ( $z_{min}$  et  $z_{max}$ ) ainsi que les attributs de la particule associée (structure **Particule**).

L'algorithme simplifié de rendu d'une particule avec le P-buffer est le suivant :

---

**Algorithme 4.3** Algorithme de rendu d'une particule avec le P-buffer.

---

```

Test de visibilité de la particule dans l'espace de la scène
Projection de la particule dans l'espace de l'observateur
Clipping de la sphère projetée par les limites de l'image
Balayage de la sphère pour conversion en pixel (<<rasterization>>)
Pour chaque pixel P issu de la conversion :
  Calculer P.Zmin profondeur minimum de la surface de la sphère
  Calculer P.O opacité de la sphère
  Soit Z la valeur présente dans le Z-buffer
  Soit O la valeur d'opacité présente dans la mémoire d'image
  Si P.Zmin > Z et O = 1
  Alors ce nouveau pixel n'est pas visible
  Sinon Si ( P.Zmin < Z et P.O = 1 ) et
    ( il n'y a pas de fragment pour ce pixel )
    Alors ce nouveau pixel est visible et cache totalement l'ancien
    Sinon on ne peut pas résoudre la visibilité
      Créer un fragment référençant cette particule
      Insérer le fragment dans la liste du pixel en fonction de P.Zmin
  FinSi
FinSi
FinPour

```

---

Le processus de compression réalise la composition des fragments et calcule la couleur, l'opacité et la profondeur effective de chaque pixel. C'est une étape qui peut être très coûteuse en calcul mais elle n'est effectuée qu'une fois le rendu de toutes les particules terminé. L'algorithme est le suivant :

L'algorithme 4.4 décrit une technique simple pour la compression des fragments. Le critère de composition repose sur l'évaluation de la surface couverte par le fragment. Un algorithme plus précis utiliserait un masque de sous-pixels décrivant la couverture du pixel. Les fragments seraient alors composés à une résolution supérieure permettant ainsi un anti-aliasing de qualité.

On peut remarquer que si le P-buffer permet de prendre en compte correctement la transparence, il est assez coûteux en calcul et en mémoire. En effet, dans le cas de scènes complexes intégrant beaucoup d'éléments semi-transparents, le nombre de fragments par pixel peut être très élevé. Il peut être ainsi nécessaire de compresser régulièrement les fragments afin de limiter la mémoire utilisée par le processus de rendu.

---

**Algorithme 4.4** Algorithme simple de compression des fragments.
 

---

```

Pour tous les pixels de l'image
  Si il existe au moins un fragment pour ce pixel
    Soit O.c = 0 l'opacité courante
    Soit C.c = (0,0,0) la couleur courante
    Tant qu'il existe un fragment et O.c < 1 Faire
      Calculer S.frag surface du pixel couverte par le fragment
      Soit O.frag l'opacité du fragment
      Soit C.frag la couleur du fragment
      O.c = O.c + (1- O.c) * S.frag * O.frag
      C.c = C.c + O.c * C.frag
    FinTantque

    Couleur finale du pixel = C.c
    Opacité finale du pixel = O.c
  FinSi
FinPour

```

---

En conclusion, la technique du P-buffer est intéressante si l'on désire un calcul exact des transparences. Dans le cas général où la majorité des particules est opaque, le P-buffer est probablement trop coûteux par rapport au Z-buffer en terme de calcul et de mémoire.

#### 4.2.6 Conclusion sur le rendu

Les choix que nous avons fait pour l'architecture du système de rendu satisfont trois objectifs principaux :

- **performance** : les techniques employées sont simples et rapides. Il est possible de rendre rapidement un nombre important de particules (environ 10000 particules par secondes sont rendues sur station SUN Ultra Sparc 1 sans accélérateur graphique).
- **qualité** : les principales fonctionnalités d'un moteur de rendu réaliste sont présentes. Cette architecture permet de traiter facilement les ombres portées, les textures des surfaces, les différents modèles de caméra, etc.
- **extensibilité** : la décomposition de l'architecture en modules indépendants chaînés par les données qu'ils émettent et reçoivent permet très facilement d'étendre un aspect du moteur de rendu. Ce peut être définir un nouveau module **réflexion de la lumière** ou étendre les possibilités de texture en complétant le module **surface**.

Cette architecture n'introduit pas de concepts très originaux mais s'avère très adaptée à notre problème de rendu d'éléments et de paysages naturels.



## Chapitre 5

# Conclusion

L'introduction des automates d'états finis permet de contrôler efficacement l'évolution d'un système de particules. Les systèmes de particules indépendantes sont en général utilisés pour modéliser et visualiser des phénomènes très peu structurés comme le feu ou l'eau [Ree83, Sim90]. Notre objectif en introduisant les automates, était de disposer d'un cadre permettant de spécifier le comportement des particules afin de pouvoir modéliser des objets ou des phénomènes structurés comme les végétaux. Les états et les transitions de l'automate permettent non seulement de définir les comportements et les changements de comportements des particules, mais aussi de décider de la mort ou de la naissance de nouvelles particules. De plus, les actions et les fonctions associées prennent en compte, d'une part l'état des particules et d'autre part la configuration de l'environnement pour décider de l'évolution de la particule.

L'implémentation d'un tel système est simple et efficace. Elle permet de gérer un très grand nombre de particules indépendantes car à aucun moment une description géométrique de la scène n'est stockée. Une technique de rendu à base de Z-buffer ou de P-buffer est suffisante et s'accorde bien avec les contraintes d'un tel système.

Dans le chapitre suivant, nous présenteront l'application de cette technique à la modélisation de végétaux. Nous verrons que les actions et les fonctions contrôlant le processus sont simples et que la topologie de la structure ramifiée engendrée par le système dépend directement de la topologie de l'automate. En conclusion, nous comparerons l'approche « automate-système de particules » aux grammairies L-systems proposées par Prusinkiewicz.



## Quatrième partie

# Applications dans le cadre des paysages naturels





# Chapitre 1

## Introduction

Dans cette partie, nous allons présenter l'application des systèmes de particules à la modélisation d'éléments naturels. Un des aspects les plus importants d'un paysage naturel est bien sûr la végétation qui couvre le relief. La première et la principale application de notre système de particules est donc dédiée à la création de la végétation. C'est un sujet très vaste comme le montre l'état de l'art à ce propos (cf. chapitre 1.1, page 17). Nous allons expliquer dans un premier chapitre comment des structures simples ou complexes de végétaux peuvent être modélisées par des particules et des automates. Nous présenterons de nombreux exemples et des extensions du principe de base pour la simulation de la croissance ou l'intégration d'influences externes.

Une autre utilisation de nos systèmes de particules est présentée dans le chapitre suivant : la visualisation de nuages. Nous montrons que des réseaux de particules semi-transparentes peuvent créer des images convaincantes bien que la structure générée n'ai aucun rapport avec une topologie sous-jacente du nuage (d'ailleurs, cette topologie existe-t-elle?).



## Chapitre 2

# La végétation

### 2.1 Introduction

La modélisation et le rendu de végétaux à l'aide de systèmes de particules ont déjà été réalisés par Reeves en 1985 [RB85]. Cependant, les arbres produits par Reeves ont un caractère « impressionniste », l'impact visuel étant principalement créé par le rendu probabiliste des feuilles. Nous utilisons également les systèmes de particules et le rendu probabiliste mais le contrôle de l'automate nous permet de définir précisément la structure ramifiée de la plante. En effet, cette structure ramifiée est caractéristique de l'espèce du végétal considéré, il est donc important de pouvoir reproduire cette topologie.

La forme des branches est représentée par la trajectoire des particules, tandis que la création des branchements résulte de la naissance de nouvelles particules. Au niveau de l'automate, les états et les actions associées ont donc le contrôle des branches alors que les transitions et les fonctions décident des branchements.

Dans les paragraphes suivants nous détaillons le principe de la modélisation des végétaux. Nous verrons également qu'il est possible de simuler une impression de croissance ou d'intégrer dans le processus de modélisation des influences externes comme celle du vent, du soleil ou du voisinage. Le rendu étant un facteur important pour le réalisme, nous parlerons dans un chapitre des techniques employées pour rendre les particules. Enfin, nous présenterons les résultats obtenus et discuterons de l'intérêt de cette méthode.

### 2.2 Principe

Nous avons défini précédemment trois étapes dans la spécification d'un système de particules contrôlé par automate (cf. chapitre 3, page 81). Nous allons maintenant répondre à ces questions dans le cas des végétaux.

## Que vont représenter les particules et leurs comportements par rapport au végétal?

Les particules et leurs trajectoires modélisent les branches de la plante. À chaque itération, le mouvement des particules est donc intégré dans l'espace 3D. Les attributs spécifiques principaux des particules sont la position, la taille et la direction de déplacement. La particule initiale aura une taille importante et représentera la base du tronc. Finalement, des particules plus petites modéliseront les rameaux terminaux et les fleurs ou les fruits (cf. figure 2.1).

## Comment décrire la structure de l'objet par l'automate d'états finis?

La structure ramifiée du végétal est représentée dans l'automate par le principe suivant :

- un état représente un type de comportement : c'est l'action associée à un état qui contrôle la trajectoire d'une particule. Il s'agit principalement d'intégrer diverses contributions pouvant influencer la forme de la branche (gravité, tropisme). Cette action modifie également d'autres attributs de la particule comme la taille ou l'âge (la taille décroît et l'âge va augmenter).
- une transition représente un branchement : la fonction associée à la transition décide de créer une nouvelle particule pour modéliser une nouvelle branche. Cette fonction calcule les attributs initiaux de cette nouvelle particule en fonction des attributs de l'ancienne particule et de paramètres spécifiques associés à la transition. C'est uniquement au moment de la mort d'une particule (quand elle atteint son âge maximum) qu'une transition va être activée et créer un branchement.

Il y a donc une relation directe entre la topologie de l'automate et la structure ramifiée modélisée. Les actions et les fonctions ont pour objectifs de modifier ou d'initialiser les attributs des particules (cf. figure 2.2).

## Comment rendre les particules pour obtenir une image convainquante et réaliste de l'objet?

Le rendu des particules dépend de ce qu'elles représentent : la particule modélisant une branche par sa trajectoire sera une sphère tandis que la particule modélisant une feuille sera un polygone. Ainsi le choix d'une forme pour une particule dépend du *type* de particule. Dans une grande majorité des cas, les particules seront représentées sous forme de sphères, les caractéristiques de rendu de ces sphères dépendant directement des attributs des particules (une primitive de cône généralisé est également disponible pour joindre deux sphères). Par exemple, les grosses particules auront une texture d'écorce (le tronc), les plus petites seront d'une couleur foncée (les branches) et enfin les feuilles seront vertes (cf. figure 2.3).

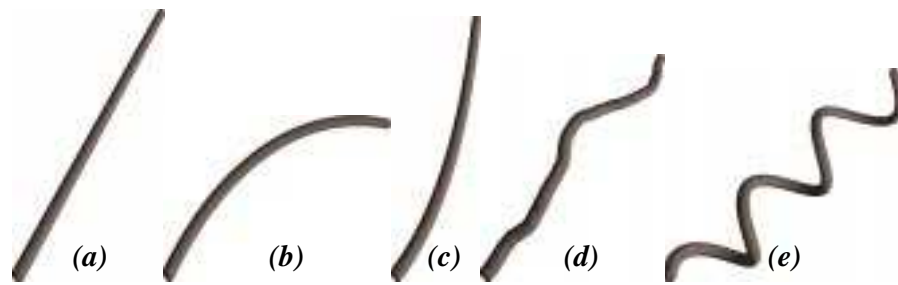


FIG. 2.1 – *Différents types de trajectoire : (a) rectiligne, (b) plagiotropique, (c) orthotropique, (d) perturbée, (e) spirale.*

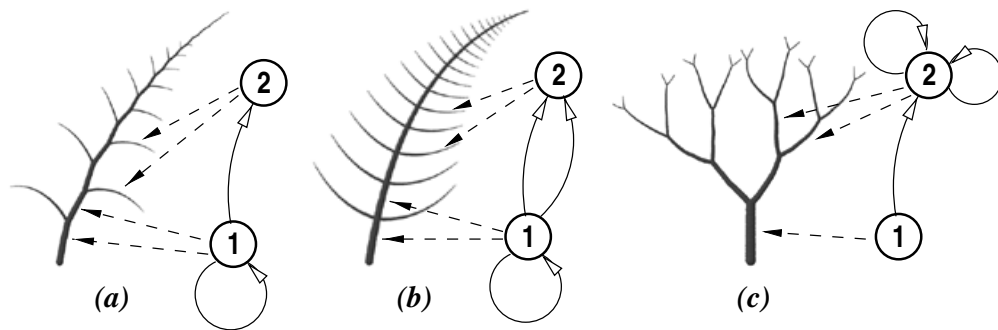


FIG. 2.2 – *Différents types de structures ramifiées et les automates associés.*

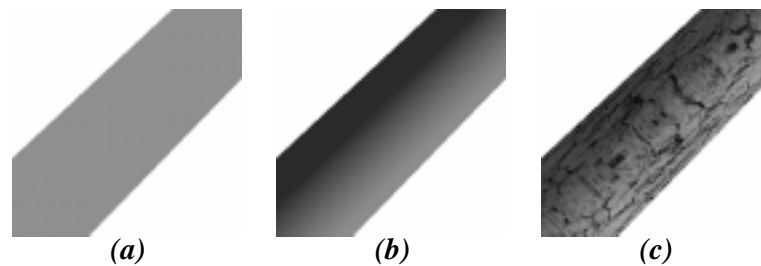


FIG. 2.3 – *Différents types de shading pour une branche : (a) constant, (b) réflexion lambertienne, (c) texture d'écorce sur la surface.*

## 2.2.1 Les attributs des particules et les caractéristiques des actions et fonctions

### 2.2.1.1 Les attributs des particules

Comme nous l'avons vu au paragraphe 4.1.1 (page 86), les attributs des particules peuvent être classés en deux catégories : les attributs standards nécessaires à la gestion de la particule et les attributs spécifiques dépendant de la modélisation. Pour la création de végétaux, nous avons défini les attributs spécifiques suivants :

- P.position : la position de la particule;

- P.direction : la direction de déplacement de la particule;
- P.plan : un vecteur perpendiculaire à la direction (définit un repère orthogonal dans l'espace avec P.position et P.direction);
- P.taille : la taille de la particule (interprétée comme le diamètre de la sphère);
- P.age : l'âge courant de la particule;
- P.age\_max : l'âge maximum de la particule;
- P.type : le type de la particule (en général branche ou feuille).

La valeur de ces attributs évolue en fonction du comportement défini par les actions associées aux états, sauf pour les attributs P.age\_max et P.type initialisés à la naissance de la particule et qui restent constants.

### 2.2.1.2 États et actions

Les actions définissent le comportement des particules à chaque itération du système. Dans notre application de modélisation des végétaux, les actions sont responsables de la forme des branches, que ce soit l'évolution de leur diamètre ou la tendance de croissance. Un certain nombre de valeurs associées à l'état paramétrisent les actions :

- E.gravité : l'influence de la gravité sur la trajectoire des particules (pour simuler une croissance plagiotropique);
- E.croissance : l'influence de l'ensoleillement (pour simuler une croissance orthotropique);
- E.perturbation : l'influence d'un facteur aléatoire perturbant la trajectoire;
- E.variation\_taille : la variation de la taille de la particule;
- E.taille\_min : la taille minimum de la particule.

Un traitement unique est réalisé pour toutes les actions de l'automate (cf. algorithme 2.1). Ce traitement prend en compte les données spécifiques de l'état définies ci-dessus et les attributs de la particule.

La valeur de Delta définit le pas d'intégration de la trajectoire des particules.

### 2.2.1.3 Transitions et fonctions

Lors de la modélisation, la création de nouvelles ramifications est représentée par les transitions entre les états de l'automate. Quand une particule atteint son âge maximum, elle peut donner naissance à des nouvelles particules si les fonctions associées aux transitions

---

**Algorithme 2.1** Algorithme des actions associées aux états.

---

```

Si P.age >= P.age_max ou P.taille < E.taille_min
  Alors supprimer la particule
Si E.gravité <> 0 Alors
  Appliquer à P.direction une force verticale de haut en bas
  de module E.gravité
Si E.croissance <> 0 Alors
  Appliquer à P.direction une force verticale de bas en haut
  de module E.croissance
Si E.perturbation <> 0 Alors
  Appliquer une perturbation aléatoire à P.direction
  d'amplitude E.perturbation

Si il existe des influences externes Alors
  Modifier P.direction          (cf. section 2.4, page 110)

P.position = P.position + P.direction * Delta
P.taille = P.taille - Delta * E.variation_taille
P.age = P.age + Delta

```

---

autorisent cette création. Le choix de créer ou non de nouvelles particules dépend principalement d'une probabilité associée à la transition. De plus, dans le cas d'une création, la fonction devra initialiser les attributs de la nouvelle particule. Comme pour les états et les actions, certains paramètres sont définis pour les transitions, tandis que les fonctions réalisent le même traitement quel que soit le type de transition. Les paramètres des transitions sont les suivants :

- T.proba\_min et T.proba\_max : intervalle de probabilité définissant la validité de la transition;
- T.rap\_taille et T.var\_rap\_taille : rapport des tailles entre la nouvelle et l'ancienne particule, ainsi qu'une valeur de dispersion;
- T.rap\_age\_max et T.var\_rap\_age\_max : rapport des âges maximums entre la nouvelle et l'ancienne particule, ainsi qu'une valeur de dispersion;
- T.alpha et T.var\_alpha, T.beta et T.var\_beta : angles de rotation du repère de la nouvelle particule par rapport au repère de l'ancienne et les valeurs de dispersion associées (cf. figure 2.4);
- T.type : type de la nouvelle particule.

Chaque paramètre intervenant dans le calcul des attributs de la nouvelle particule est donné avec un intervalle de dispersion. En effet, des facteurs stochastiques sont nécessaires



pour créer une infinité de modèles différents à partir de la définition des caractéristiques de l'espèce. L'algorithme 2.2 des fonctions est exécuté à la mort de la particule, c'est à dire quand la condition  $P.age \geq P.age\_max$  est réalisée.

---

**Algorithme 2.2** Algorithme des fonctions associées aux transitions pour le contrôle des ramifications.

---

```

Si ( T.proba_min <= valeur aléatoire < T.proba_max) Alors
  Créer une nouvelle particule Pnouv

  Pnouv.position = P.position
  Pnouv.direction = P.direction
  Pnouv.plan = P.plane

  Beta = AleaUniforme( T.beta , T.var_beta )
  Appliquer à Pnouv.plan une rotation d'axe P.direction et d'angle Beta
  Alpha = AleaUniforme( T.alpha , T.var_alpha )
  Appliquer à Pnouv.direction une rotation d'axe Pnouv.plan et d'angle Alpha

  Facteur_age_max = AleaUniforme( T.rap_age_max , T.var_rap_age_max )
  Pnouv.age = P.age * Facteur_age_max

  Facteur_taille = AleaUniforme( T.rap_taille , T.var_rap_taille )
  Pnouv.taille = P.taille * Facteur_taille

  Pnouv.age = 0
  Pnouv.type = T.type

  L'état courant de Pnouv devient l'état destination de la transition
FinSi

```

---

La fonction `AleaUniforme( val_moy , écart )` retourne une valeur aléatoire équiprobable dans l'intervalle  $[ val\_moy - \text{écart} , val\_moy + \text{écart} ]$ .

## 2.3 Premiers exemples

La valeur des paramètres associés aux états et aux transitions sont déduits de la morphologie de la plante à modéliser. C'est l'observation de la structure ramifiée d'un végétal qui indique la topologie de l'automate, les angles de branchement, la forme des branches ou le placement des feuilles. Des automates simples permettent déjà de construire des structures complexes (cf. figures 2.5 et 2.6). Dans tous ces modèles pour la création de végétaux, une seule particule initiale est créée à la base du tronc. Une particule est supprimée quand sa taille devient inférieure à un minimum fixé par un paramètre de l'état ou si son âge

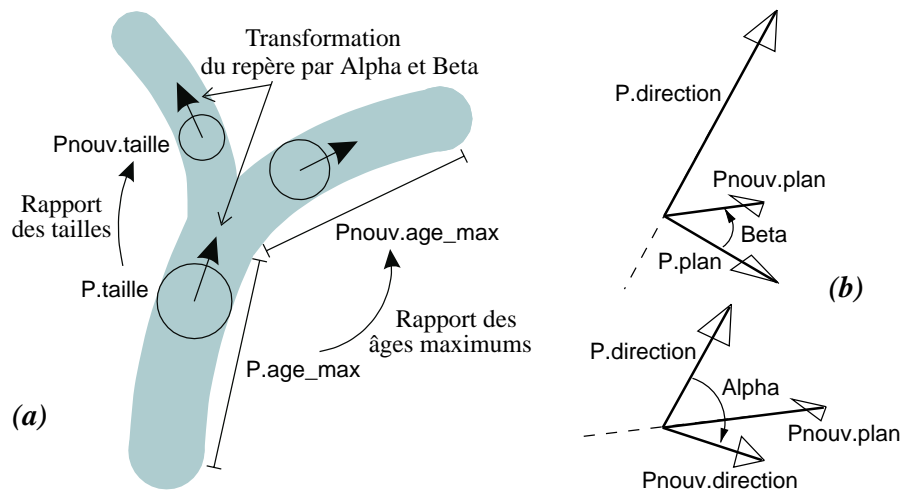


FIG. 2.4 – Les attributs des nouvelles particules : (a) les attributs des nouvelles particules dépendent de ceux de la particule mère, (b) modification par Alpha et Beta du repère des particules.

courant devient égal à son âge maximum (cf. algorithme 2.1). Dans ce dernier cas, les transitions partant de l'état pourront donner naissance à de nouvelles particules. L'évolution du système est terminée quand il n'existe plus aucune particule.

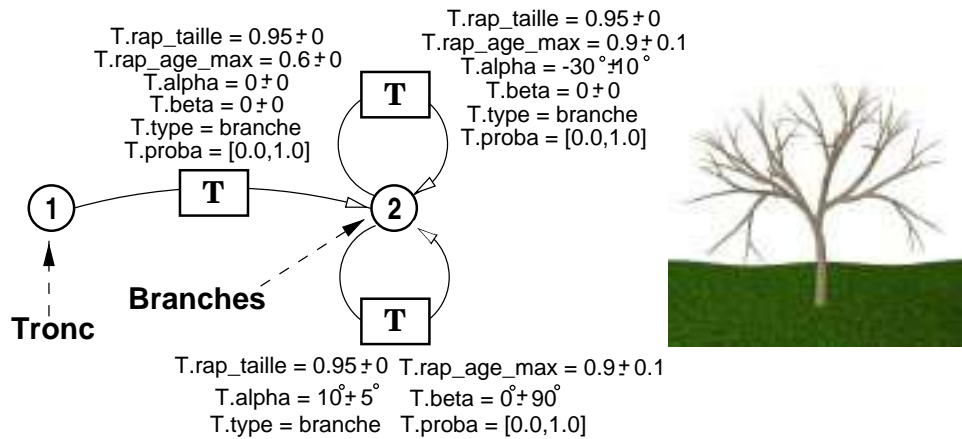


FIG. 2.5 – Un exemple d'automate et l'image de la structure ramifiée modélisée. Les paramètres associés à chaque transition sont indiqués.

Comme nous le voyons dans les exemples ci-dessus, certaines valeurs de paramètres sont données avec une tolérance (par exemple  $T.ALPHA = -30^\circ \pm 10^\circ$ ). Un générateur de nombres aléatoires est donc utilisé pour choisir de façon équiprobable une valeur dans cet intervalle. Nous utilisons une suite aléatoire de type  $u_n = f(u_{n-1})$ , la valeur de  $u_0$  étant appelée *graine*. Pour une graine donnée, le système produira un exemplaire unique du végétal. La graine est donc **le seul paramètre à mémoriser pour pouvoir reproduire exactement**

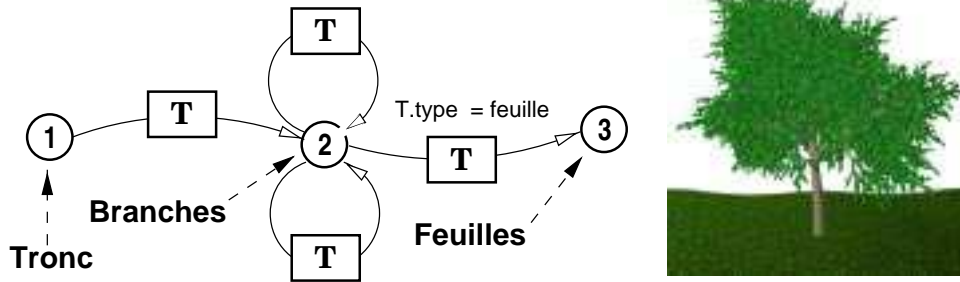


FIG. 2.6 – Le même automate avec un état supplémentaire. Les particules dans cet état représentent les feuilles de l'arbre.

le même modèle d'une espèce (la définition de l'espèce étant contenue dans la définition de l'automate). C'est un avantage important de cette méthode, les données nécessaires à la définition d'un objet étant ainsi très limitées.

## 2.4 Évolution et Influences externes

Il est possible avec ce système de simuler visuellement un certain nombre de phénomènes. Il est important ici de rappeler un de nos objectifs pour ce système de rendu de paysages naturels : nous considérons un modèle pour la création d'éléments naturels comme satisfaisant si l'image générée atteint un **réalisme visuel** correct. Ainsi, les méthodes développées dans la suite sont essentiellement des approches empiriques de problèmes très complexes. Cependant, elles remplissent parfaitement leurs rôles car elles permettent de simuler visuellement des effets naturels de façon convainquante et à moindre coût.

### 2.4.1 La croissance

Bien que notre système pour la modélisation des végétaux ne soit pas un moteur de croissance, il peut être intéressant de visualiser un effet de croissance pour les végétaux. Cette impression de croissance est simplement réalisée en choisissant des valeurs initiales variables pour les attributs de la première particule (celle représentant le tronc). En effet, les attributs des nouvelles particules dépendent des attributs de la particule dont elles sont issues (les caractéristiques d'une branche d'ordre  $i$  dépend des caractéristiques de la branche d'ordre  $i - 1$ ). Ainsi, toute la structure d'un végétal est fonction des attributs de la première particule.

Pour modéliser des végétaux à plusieurs niveaux de croissance, ce sont les attributs `P.taille` et `P.age_max` qui changent. L'attribut `P.taille` représente le diamètre du tronc et donc l'épaisseur des branches. L'attribut `P.age_max` donne l'espérance de vie de la particule et donc la longueur du tronc et des branches sous-jacentes. La figure 2.7 montre un exemple

de simulation de croissance par modification linéaire de ces attributs.



FIG. 2.7 – *Simulation visuelle de la croissance d'un arbre (temps de calcul inférieur à 10 secondes sur station de travail SUN Ultra Sparc 1, sans accélérateur graphique).*

### 2.4.2 Les contraintes spatiales

Contraindre la structure d'un végétal à rester dans des limites de l'espace 3D peut être simplement réalisé avec notre technique basée sur les systèmes de particules. D'une part, il n'y a pas de phase d'interprétation entre la description de la structure et l'espace géométrique (comme dans les modèles à base de L-systems, cf. section 1.1.1.5, page 26). À tout moment de son évolution, une particule connaît sa position dans l'espace et son interaction avec l'environnement. D'autre part, la trajectoire et les attributs des particules sont contrôlés par les actions associées aux états. Il est facile d'intégrer dans cette action des contraintes ou des influences qui modifient la trajectoire normale de la particule (cf. figures 2.8 et 2.9).



FIG. 2.8 – *Action d'une sphère sur la trajectoire des particules (et donc sur la forme des branches du buisson). Le temps de calcul de ces images est de 6 secondes, avec ou sans contraintes (voir aussi planches couleurs page 174).*

### 2.4.3 Les contraintes spatiales et temporelles

Certains phénomènes ont des effets immédiats sur la géométrie de la plante et n'influent pas seulement sa croissance. Dans le cas de production d'animations, les effets du vent par exemple doivent être pris en compte. La forme des branches dépend donc de la structure de



FIG. 2.9 – *Contraintes et phototropisme : (a) deux vues d'un végétal contraint par deux plans parallèles, (b) un végétal sans et avec l'influence d'une source de lumière.*

l'arbre et des attributs des particules (contraintes spatiales) mais aussi de la force exercée par le vent à un instant donné (contrainte temporelle).

Pour recréer cet effet, la trajectoire de chaque particule va être modifiée pour que la branche modélisée oscille à une fréquence propre. Une fonction sinusoïdale est utilisée comme génératrice des mouvements d'oscillation. La fréquence et l'amplitude des oscillations sont fonctions de la taille des particules (le diamètre des branches). Ainsi, l'amplitude de la force appliquée à une particule (en plus des autres influences) sera :

$$F_{vent} = f(P.taille, ENV.temps\_global, ENV.vitesse\_du\_vent)$$

avec  $P.taille$  représentant la taille courante de la particule,  $ENV.temps\_global$  et  $ENV.vitesse\_du\_vent$  définissant le temps global de la scène et la vitesse du vent. Ces deux dernières variables font partie de la définition de l'environnement de la scène (cf. Un système pour la synthèse de paysages, partie VI, page 153). La direction du vecteur  $\vec{F}_{vent}$  est donnée par le paramètre  $ENV.direction\_du\_vent$  (cf. figure 2.10).



FIG. 2.10 – *Influence du vent sur arbre. Cet arbre est composé de 220000 particules, le temps de modélisation et de rendu est de 20 secondes (voir aussi planches couleurs page 174).*

Dans le cas d'animations à grande échelle, comme le mouvement d'une prairie dans le vent, les paramètres de l'environnement  $ENV.vitesse\_du\_vent$  et  $ENV.direction\_du\_vent$  pour-

ront être fonction de l'espace 3D et du temps. Une formulation basée sur la propagation des ondes peut servir à exprimer la direction et l'amplitude de  $\vec{F}_{vent}$ .

## 2.5 Rendu

### 2.5.1 Fonction d'illumination probabiliste

Une grande partie du réalisme visuel d'un élément naturel dépend de la qualité du rendu et en particulier de la prise en compte de l'illumination. La complexité géométrique de ces éléments et la technique de rendu employée (le Z-buffer) nous interdisent l'utilisation de méthodes « exactes » pour le calcul des contributions lumineuses. Ainsi, il n'est pas possible de lancer des rayons en direction d'une source lumineuse pour tester sa visibilité. Il n'est pas raisonnable non plus d'utiliser les cartes d'ombre (cf. chapitre 2.1.1, page 66), celles-ci étant trop peu précises pour déterminer l'éclairage sur une particule individuelle d'un arbre. Le calcul d'illumination probabiliste introduit par Reeves [RB85] est donc l'alternative choisie. Une fonction d'illumination probabiliste détermine une probabilité d'éclairage d'un point par une source. Pour ce faire, cette fonction utilise des variables donnant la position de la particule par rapport au volume de l'objet (cf. figure 2.11).

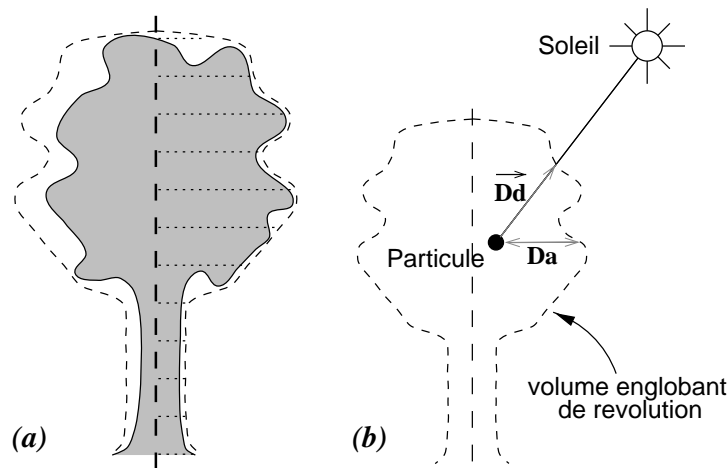


FIG. 2.11 – Fonction d'illumination probabiliste : (a) lors de la construction de l'objet, un volume englobant de révolution est maintenu, (b)  $D_a$  et  $D_d$  sont les plus petites distances de la particule au volume englobant. Le surcoût en calcul impliqué par la fonction d'illumination probabiliste est inférieur à 1 % du coût total.

Le calcul de l'illumination finale pour le rendu d'une particule est donc :

$$K_a = f_a(D_a)$$

$$K_d = f_d(\vec{D}_d)$$

$$I_{finale} = Random() * 0.2 + (SURF.ambient * K_a) + (SURF.diffus * K_d)$$

avec  $f_a$  et  $f_b$  des fonctions aléatoires simples pour le calcul de la composante ambiante ( $K_a$ ) et de la composante diffuse ( $K_d$ ) de l'illumination. La fonction  $Random()$  tire un nombre aléatoire dans  $[0, 1]$  tandis que  $SURF.ambient$  et  $SURF.diffus$  sont des attributs de la surface représentant le coefficient de réflexion pour la lumière ambiante et la lumière diffuse.

Cette méthode probabiliste pour le calcul de l'illumination peut être aussi utilisée dans le cas de l'herbe et des nuages (cf. section 3.3, page 123). Il suffit d'obtenir deux variables  $D_a$  et  $D_d$  qui définissent une probabilité d'éclairage par la lumière ambiante ou la lumière directe (celle du soleil). Dans le cas du gazon et des nuages elles sont définies comme des distances de la particule à un volume englobant (cf. figure 2.12).

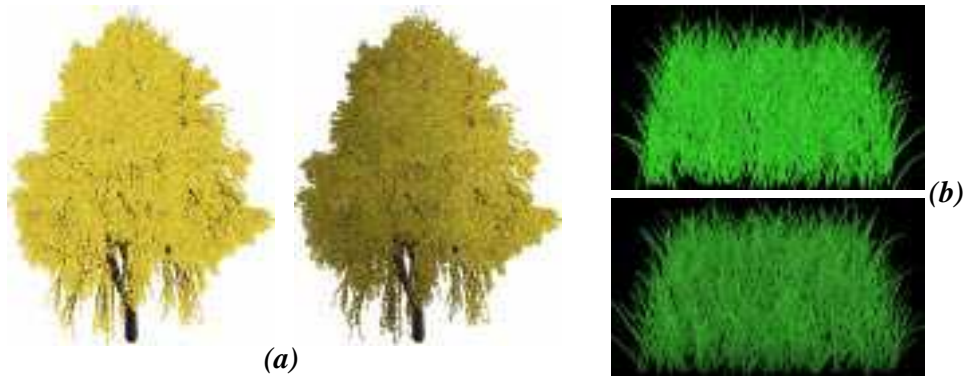


FIG. 2.12 – Effets de l'illumination probabiliste : (a) un arbre sans et avec le calcul de l'illumination, (b) une parcelle d'herbres sans et avec le calcul de l'illumination (voir aussi planches couleurs page 174).

### 2.5.2 Formes et textures des particules

Quand l'observateur est proche du végétal, il peut distinguer de nombreux détails dans la structure et l'apparence de la plante. Les modèles "impressionnistes" pour la synthèse d'éléments naturels se comportent en général assez mal dans de telles situations. Pour les systèmes de particules en particulier, le réalisme est créé par une multitude de particules simples et non par une géométrie complexe et détaillée.

Nous proposons un certain nombre de solutions dans le cas où une modélisation fine et réaliste est requise :

- la géométrie des embranchements : c'est un problème difficile à traiter dans le cas de surfaces polygonales. Bloomenthal propose une solution basée sur les surfaces splines [Blo85] qui donne de très bons résultats. Néanmoins, dans notre modèle à base de

particules, les branches sont uniquement représentées par des successions de sphères connectées. Il est possible d'améliorer le réalisme des embranchements en imposant une transition progressive et continue dans la taille des particules (quand une particule fille est créée).

- l'apparence des branches : s'il est possible de négliger la texture des branches dans le cas d'un arbre lointain, il est nécessaire dans une vue rapprochée d'intégrer le dessin de l'écorce au rendu des branches. Il existe essentiellement deux méthodes pour modifier l'aspect d'une surface : le plaquage d'images 2D et la texture procédurale. Le plaquage de texture peut donner un résultat très réaliste mais se heurte à de nombreux problèmes : calcul des coordonnées de texture, continuité du motif ou problèmes d'aliasing. Nous avons choisi d'utiliser une texture procédurale définie comme une fonction de l'espace 3D qui perturbe la normale à la surface des particules pour créer un effet d'écorce.
- la géométrie des feuilles, fleurs et fruits : quand l'observateur est près du végétal, il est indispensable de représenter ces attributs de façon plus détaillée qu'une simple sphère. Dans ce cas, la forme de la particule pourrait être représentée par un objet polygonal ou pourquoi pas, par un autre système de particules.

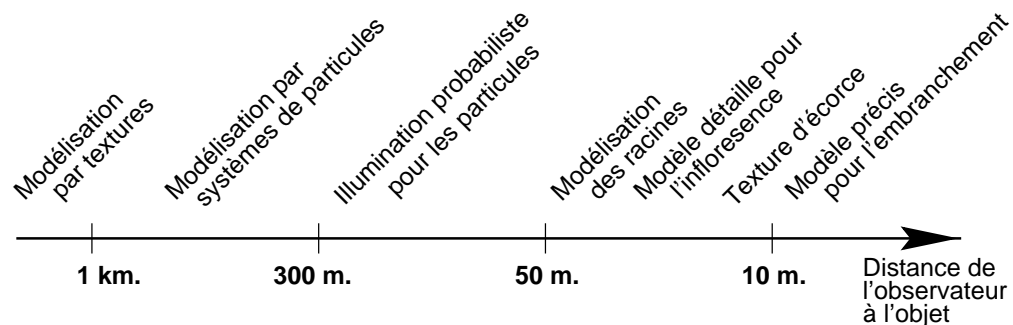


FIG. 2.13 – Niveau de détail de la modélisation en fonction de la distance.

Toutes ces méthodes permettant d'accroître le niveau de détail des végétaux peuvent s'avérer très coûteuses. Il est donc indispensable de les utiliser lorsqu'elles sont nécessaires, c'est-à-dire quand le réalisme visuel peut être significativement augmenté. Ces préoccupations rejoignent le problème des représentations multi-échelles développé dans le chapitre 2 (page 133). La figure 2.13 donne une estimation empirique des distances à partir desquelles les méthodes d'amplification des détails doivent être employées.

## 2.6 Résultats

Dans ce paragraphe nous allons présenter quelques résultats obtenus par notre méthode de création de végétaux. En particulier, nous illustrerons la modélisation d'un objet en



présentant les différentes étapes de la définition de l'automate pour un modèle de sapin.

La construction d'un modèle de végétal consiste à spécifier la topologie de l'automate et les paramètres associés aux états et aux transitions. Une bonne approche repose sur la construction progressive des différents niveaux de la structure ramifiée du végétal. La figure 2.15 illustre la spécification d'un automate pour un modèle de sapin.

En règle générale, la construction d'un modèle pour un végétal est basée sur l'observation de sa morphologie et des caractéristiques de ses branchements. L'automate correspondant aura autant d'états que le végétal possède de niveaux de ramification. De plus, un ou des états représentent la création de feuilles, les transitions incidentes à cet état modifiant le type de particule. Il est également possible de modéliser les feuilles ou les fruits par des systèmes de particules.

Des automates simple peuvent générer des objets assez réalistes et complexes comme le montre la figure 2.14. Grâce aux tolérances données pour les paramètres des transitions, un nombre illimité d'objets différents mais semblables peut être construit.

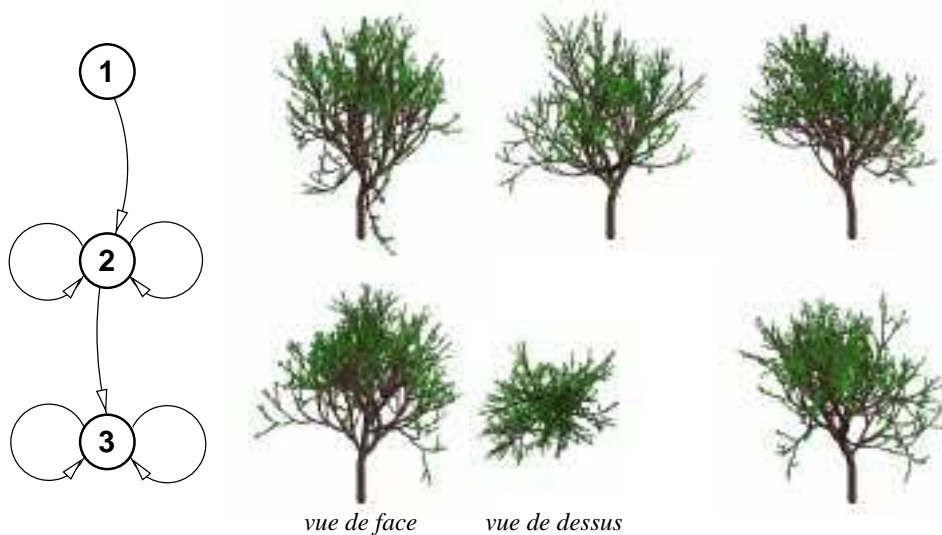


FIG. 2.14 – La structure d'un automate (hormis l'état pour les feuilles) et plusieurs arbres associés. Bien que l'automate soit simple, les objets produits sont réalistes et tous différents (voir aussi planches couleurs page 174).



## 2.7 Extensions

On peut imaginer de nombreuses applications et améliorations aux systèmes de particules contrôlés par automates. Dans cette section, nous proposerons quelques outils pour faciliter la création de modèle et envisageons d'autres utilisations à cette technique :

- **modéliser également la structure ramifiée des racines.** Pour compléter la création d'un végétal, les automates pourraient intégrer la topologie du réseau de racine. Bien sûr, la trajectoire des particules représentant les racines serait contrainte à rester dans le sol.
- **imbriquer les automates.** Au lieu de contrôler uniquement la trajectoire d'une particule par les états de l'automate, il est possible d'associer un nouvel automate à un état. Une particule atteignant cet état serait donc la particule initiale d'un nouveau système. On peut ainsi envisager la création d'une forêt avec un seul de ces automates récursifs : l'automate de plus haut niveau définissant le placement des arbres, l'automate suivant modélisant un arbre et pour finir l'automate de plus bas niveau créant une feuille.
- **réaliser le « morphing » entre deux espèces de végétaux.** Les végétaux étant complètement définis par l'automate et les données associées, on peut imaginer produire de nouvelles espèces en mélangeant par interpolation deux automates.
- **créer un outil interactif pour la définition de l'automate associé à un objet.** Cet outil faciliterait la création de la topologie de l'automate grâce à une visualisation immédiate de la structure ramifiée produite. L'utilisateur pourrait facilement modifier les paramètres associés aux états et aux transitions, mais aussi écrire de petits scripts définissant des comportements particuliers pour les particules. Ces scripts seraient interprétés au moment de la construction de l'objet.

## 2.8 Conclusion

Nous avons présenté dans ce chapitre une technique pour la création de végétaux. L'évolution d'un système de particules indépendantes va créer la structure géométrique de l'objet. Cet évolution est représentée par un comportement associé à chaque particule. Pour d'une part, définir des comportements différents, et d'autre part, décider des changements ou de l'affectation des comportements aux nouvelles particules, nous utilisons le formalisme des automates d'états finis. Ainsi, les états et l'action associée contrôlent la trajectoire des particules et donc la forme des branches. Les transitions, quant à elles, décident de créer de nouvelles particules qui représenteront de nouvelles branches. Le rendu de cette évolution est simplement la projection des positions successives des particules. Pour accroître le réalisme de ce modèle, des méthodes probabilistes permettent un rendu de l'éclairage sur les particules.

Nous avons défini en introduction certains choix qui devaient guider la conception d'un système de modélisation et de rendu de végétaux. Ces critères étaient les suivants : la modélisation doit s'appuyer sur des paramètres morphologiques des plantes; elle doit pouvoir intégrer des influences externes; la modélisation et le rendu doivent être simultanés; le rendu du modèle doit être réaliste; le processus complet doit être rapide. Nous pensons que le système développé dans ce chapitre remplit toutes ces conditions. Dans le tableau ci-dessous, nous donnons une évaluation de notre approche comme nous l'avons fait pour les autres modèles (cf. tableau 1.1, page 44).

Modèle	Réalisme des images	Diversité des espèces	Connaissances requises	Complexité et calcul	Influences externes
Chaudy [Cha95, Cha96]	bon	moyenne	peu	faible	oui

TAB. 2.1 – *Caractéristiques de notre technique pour la modélisation et le rendu des végétaux*

Il existe de nombreux points communs entre le formalisme des automates d'états finis et celui des grammaires. Il paraît donc intéressant de comparer notre approche avec les travaux de Lindenmayer et Prusinkiewicz sur les L-systems [PLH88].

Les L-systems expriment sous forme de règles de substitution la croissance d'une plante. C'est un processus itératif qui va produire une chaîne de symboles représentant la structure ramifiée de la plante mais aussi ses attributs. Une phase d'interprétation géométrique va générer une description 3D de l'objet qui sera ensuite rendue par des méthodes classiques. Bien que cette approche et la notre utilisent à la base des formalismes voisins, elles sont cependant différentes sur certains points :

- un L-system est un moteur de croissance. Il tente de simuler le plus fidèlement possible le processus de développement du végétal. Ce n'est pas du tout le cas de notre approche, qui s'intéresse à la modélisation d'une structure topologiquement satisfaisante en terme d'impact visuel. Par exemple, nous générons directement un arbre adulte, sans connaissance de son évolution. Les L-systems ont besoin de « faire pousser » le végétal jusqu'au stade de développement désiré.
- la création d'un objet par systèmes de particules est un processus continu, aucune représentation intermédiaire n'étant nécessaire depuis la spécification de l'automate jusqu'à l'image finale. Ce n'est pas le cas du processus de modélisation des L-systems. Notre approche permet ainsi de représenter visuellement des contraintes spatiales sur la topologie de la plante.
- du fait du processus de construction, les objets que nous générons ne peuvent porter les stigmates d'événements qui ont affecté leur croissance (accident, taille, répétitions). De plus, la forme des branches dépendant directement de la trajectoire des particules, la structure résultante de la plante ne peut être mémorisée. Il est donc impossible

d'appliquer, autrement que par des influences empiriques et locales, des contraintes sur cette structure.

Les différences entre ces deux méthodes sont caractéristiques de deux grands types d'approches : les techniques dites « empiriques » et les techniques d'inspirations physiques. Ainsi, notre modèle est loin d'être aussi précis que celui proposé par Prusinkiewicz, mais il permet, de part ces caractéristiques, une utilisation efficace dans un contexte précis : la création de paysages de synthèse.

## Chapitre 3

# Une autre application : les nuages

### 3.1 Introduction

Les nuages sont aussi des éléments importants des paysages. En plus de donner son apparence au ciel, ils ont une influence indirecte sur l'éclairage de la scène. Nous présentons dans ce chapitre l'application de nos systèmes de particules à la modélisation et au rendu de nuages.

Il faut tout d'abord remarquer que les nuages sont des éléments assez différents des végétaux ou des objets habituellement manipulés en image de synthèse. En effet, ils n'ont pas de surface bien définie et possèdent un caractère dynamique complexe difficile à modéliser. De plus, ils interagissent avec la lumière par un phénomène de diffusion au niveau de particules microscopiques. Nous avons vu dans l'état de l'art (cf. section 1.3, page 56) certains modèles pour la modélisation et le rendu de nuages ou de phénomènes similaires. Ces méthodes de simulation sont souvent complexes et très coûteuses en terme de calcul. Ainsi, de nombreux modèles empiriques existent pour représenter les nuages, avec plus ou moins de réalisme. Des résultats convaincants ont été obtenus par Gardner en utilisant des amas d'ellipsoïdes et des fonctions fractales pour la transparence (cf. section 1.3.1.2, page 59) [Gar85].

La méthode que nous avons développée entre dans la catégorie des modèles empiriques. Elle permet, d'une part de modéliser des nuages à l'aide d'automates et d'autre part de rendre les particules en tenant compte de l'illumination.

### 3.2 Des particules et des automates pour les nuages

Bien qu'il n'existe pas dans les nuages de structure ramifiée comme dans les végétaux, nous utilisons exactement le même principe. À partir d'une particule initiale appelée germe, l'ensemble des particules va croître en fonction de l'évolution imposée par un automate d'états finis. L'utilisation intensive de fonctions stochastiques permet de générer des structures très irrégulières. La différence majeure avec les systèmes de particules utilisés pour

les végétaux se situe au niveau du **type** des particules. Pour les nuages, les particules sont représentées par des sphères uniformes semi-transparentes. Finalement, un nuage est donc formé par un amas « faiblement structuré<sup>1</sup> » de particules semi-transparentes<sup>2</sup>.

Les automates pour le contrôle de ces systèmes sont simples : chaque particule doit donner naissance à une ou plusieurs particules dans une direction privilégiée. La figure 3.1 donne un exemple d'une telle construction. Les paramètres alpha et beta associés aux transitions permettent d'influencer le développement du système de particules et donc de générer des nuages avec des formes caractéristiques.

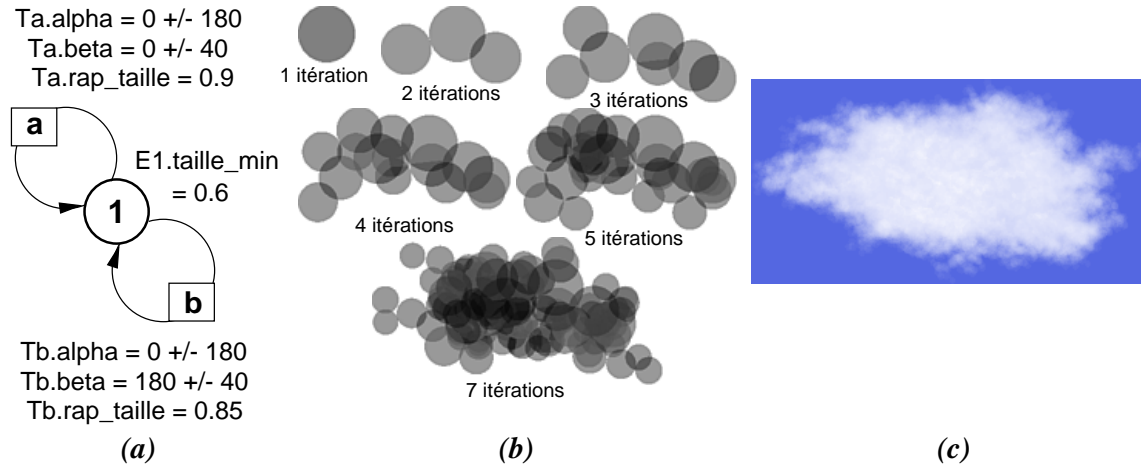


FIG. 3.1 – Un exemple d'automate pour la création de nuages : (a) la topologie de l'automate, chaque particule donne naissance à deux nouvelles particules (pour la signification des angles alpha et beta voir figure 2.4, page 109), (b) les sept premières itérations du système, (c) le nuage final est formé de 63145 particules calculées et rendues en 14 secondes.

Les paramètres de la première particule, à savoir sa taille, sa position et le repère associé, ont également une influence sur la forme et l'apparence de l'élément produit. La figure 3.2 montrent quelques nuages produits avec le même automate pour des valeurs initiales et des paramètres différents.

Comme dans le cas des végétaux, il est possible d'intégrer des influences externes en modifiant la position ou d'autres attributs des particules (cf. figure 3.3). Il est également possible de modifier les paramètres de l'automate pour produire des séquences animées de déformations de nuage.

La topologie de l'automate et les paramètres associés définissent donc un type de nuage

1. Bien qu'il n'existe pas de topologie stricte dans la forme d'un nuage, on peut cependant distinguer des types de nuage par leur apparence (cumulus, strato-cumulus, cirrocumulus ...). Ainsi chaque catégorie présente certaines caractéristiques visuelles que l'on peut interpréter comme une structure.

2. On remarque ici l'analogie entre la définition physique du nuage et le modèle proposé : un nuage est un ensemble de particules d'eau très fines, liquides ou solides, maintenues en suspension dans l'atmosphère. Mais bien sûr, les particules que nous utilisons ne représentent pas la composition d'un « vrai » nuage.

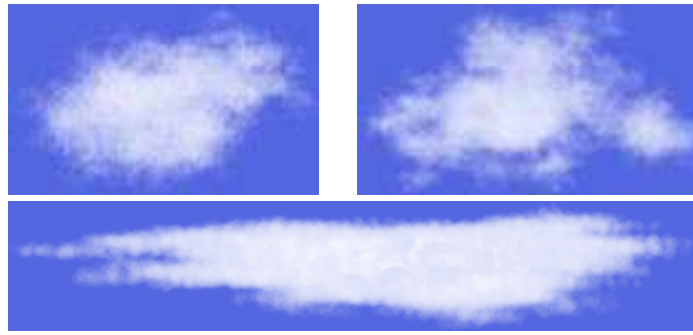


FIG. 3.2 – Trois nuages produits avec le même automate pour des paramètres différents.



FIG. 3.3 – Animation et déformation d'un nuage par un champ de forces tangentielles.

(comme ils définissaient une espèce de plante pour les végétaux). Il est possible de créer des nuages denses et compacts comme les cumulus ou bien les formes plus étirées des cirrus. Nous allons voir maintenant qu'il est possible d'intégrer empiriquement l'influence de l'illumination dans le rendu de ces éléments.

### 3.3 Le rendu des nuages

L'interaction entre la lumière et un nuage est un phénomène complexe comme nous l'avons vu dans l'état de l'art (cf. section 1.3, page 56). Cependant, des méthodes empiriques pour le calcul de la couleur du nuage peuvent donner de bons résultats. C'est-à-dire, des images qui paraissent réalistes dans les conditions d'éclairage de la scène.

Notre technique de rendu d'un nuage va calculer, pour toutes les particules, deux coefficients : le premier noté  $K_a$  représente les contributions de l'éclairage ambiant<sup>3</sup>, le second noté  $K_d$  représente l'apport de l'éclairage direct<sup>4</sup>. Comme pour la fonction d'illumination probabiliste des feuilles d'un arbre, ce sont les distances de la particule au volume englobant qui définissent ces coefficients. Nous appliquons les formules suivantes :

3. Dans le cas des nuages, l'éclairage ambiant provient de la diffusion de la lumière dans l'atmosphère, de la réflexion de la lumière par le sol et par les autres éléments du ciel.

4. Toujours dans le cas des scènes naturelles, on considère le soleil comme la seule source d'éclairage direct.



$$K_a = f_a(D_a)$$

$$K_d = f_d(\vec{D}_d)$$

$$I_{réfléchié} = (K_i + P.ambient * K_a) * P.couleur * I_a + (P.direct * K_d) * P.couleur * I_d$$

avec  $D_a$  la plus petite distance de la particule au volume englobant,  $\vec{D}_d$  la plus petite distance au volume englobant dans la direction de la source de lumière et  $f_a, f_d$  des fonctions stochastiques simples. La couleur réfléchié  $I_{réfléchié}$  dépend des caractéristiques des particules du nuage (P.couleur la couleur de la particule, P.ambient et P.direct les coefficients de diffusion de la lumière), de l'éclairage de la scène ( $I_a$  et  $I_d$  pour les contributions indirectes et directes) et de la lumière incidente sur cette particule ( $K_i, K_a$  et  $K_d$ ). Nous avons introduit la constante  $K_i$  pour représenter l'apport de la diffusion multiple par les particules à l'intérieur du nuage. La figure 3.4 présente des exemples d'illumination de nuages calculés selon ce principe.

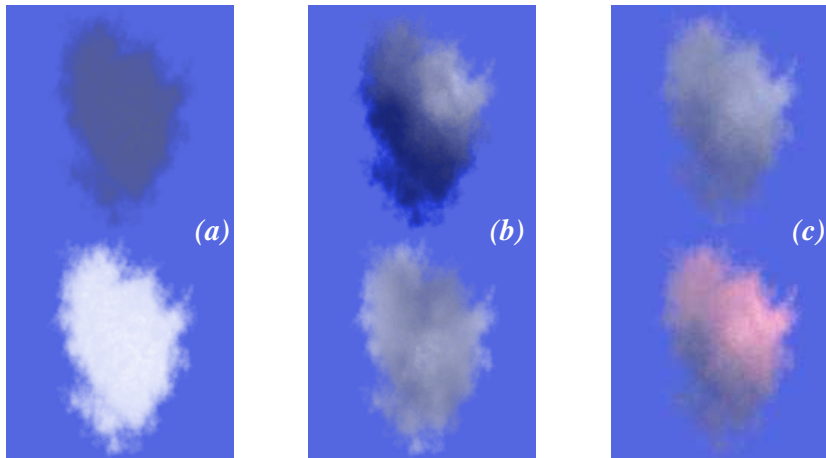


FIG. 3.4 – Le rendu des nuages : (a) le choix de la couleur des particules permet de simuler un albédo différent, (b) la contribution de l'illumination directe (en haut), la contribution de l'illumination indirecte (en bas), (c) le modèle complet avec un éclairage blanc (en haut) ou rosé au coucher du soleil (en bas). Les temps de calcul de ces images (en résolution  $500 \times 500$ ) sont compris entre 10 et 15 secondes sur station SUN Ultra Sparc 1 (voir aussi planches couleurs page 174).

### 3.4 Conclusion

Nous avons décrit un modèle pour le rendu de nuages en utilisant le formalisme des systèmes de particules contrôlés par un automate. Nous jugeons le réalisme visuel des nuages produits tout à fait satisfaisant. De plus, le coût en calcul est faible (au total, une dizaine de secondes pour le modèle complet) par rapport à des méthodes plus complexes mais plus

précises. En plus de la prise en compte de l'illumination, ce modèle permet simplement et rapidement de déformer l'élément pour simuler son comportement au cours du temps.

Après la création des végétaux, cette deuxième application de la technique des systèmes de particules indépendantes permet d'en apprécier l'intérêt dans le domaine de la modélisation et du rendu des éléments naturels.

Une perspective intéressante à ce travail serait l'étude d'une extension multi-échelle à ce modèle. En effet, un nombre important de nuages doivent souvent être intégrés et le coût en calcul peut alors devenir prohibitif. Il semble possible<sup>5</sup>, en contrôlant la taille des particules et leurs transparences, de mettre au point un modèle adaptatif en fonction de la contribution de l'élément sur l'image finale.

---

5. Par rapport au végétaux, les nuages ne possèdent pas de caractéristiques structurelles fortes. Une dégradation du modèle aurait sûrement un impact visuel moindre.



## Chapitre 4

# Conclusion

Après avoir défini les systèmes de particules et les automates associés dans la partie précédente, nous avons démontré leur validité par deux applications. Il s'agit d'un modèle complet pour la création de végétaux, et aussi d'un modèle pour la représentation de nuages. Non seulement, les images de végétaux que nous avons produites sont réalistes, mais de plus il est possible d'intégrer et de simuler, sans surcoût significatif, des phénomènes naturels complexes comme la croissance ou le vent. Ainsi, bien que cette approche ait des points communs avec celle de Reeves [RB85], elle atteint un niveau de réalisme et de fonctionnalité supérieur : la structure ramifiée est détaillée, la forme des branches est réaliste et variée, le rendu est convainquant et ceci même en plan rapproché.

L'adaptation aux nuages que nous avons proposée est également une approche intéressante. Dans des conditions d'éclairage standards, le réalisme visuel atteint est du niveau des méthodes physiques complexes bien que les coûts en calcul restent limités. De plus, cette technique a de nombreux avantages par rapport aux autres méthodes empiriques pour la création de nuages : c'est un vrai modèle 3D qui s'intègre complètement à la scène, que ce soit au niveau de l'interaction avec les autres objets ou de la prise en compte du contexte d'illumination.

Enfin, l'aspect le plus intéressant des modèles proposés est l'adéquation totale de leurs caractéristiques avec les contraintes d'intégration dans un système de rendu de paysages naturels :

- **aucune connaissance spécifique n'est requise.** La modélisation d'un végétal ou d'un nuage est déduite de l'observation de sa morphologie et de son aspect.
- **l'intégration d'influences externes.** Aussi bien pour les plantes que pour les nuages, des forces peuvent modifier le comportement des particules afin de simuler une interaction.
- **la modélisation et le rendu simultanés.** Dans notre technique, les trajectoires et les attributs des particules sont immédiatement intégrés par une méthode efficace de

rendu direct, le Z-buffer.

- **un rendu réaliste.** Bien que les techniques à base de Z-buffer ne permettent pas d'intégrer facilement tous les phénomènes optiques, les fonctionnalités de notre moteur de rendu sont suffisantes dans le cadre des paysages.
- **des processus rapides.** Que ce soit pour les végétaux ou les nuages, les temps de calculs sont limités par rapport à la complexité et au réalisme des objets générés.

Nous pensons que la technique des systèmes de particules et des automates pourrait être utilisée pour la modélisation et le rendu d'autres éléments naturels. En particulier, dans le cas de phénomènes très dynamiques comme l'eau ou les conditions atmosphériques (pluie ou neige), pour lesquels les systèmes de particules indépendantes ont déjà été utilisés. Le contrôle de tels systèmes par automates d'états finis permettrait d'imposer une structure<sup>1</sup> à ces phénomènes complexes.

---

1. Des structures apparaissent dans ces phénomènes, car ils interagissent avec un environnement qui lui est structuré: le ruisseau coule sur un terrain, la neige tombe et se dépose en fonction des courants d'air et des obstacles.

## Cinquième partie

# Modélisation multi-échelle



# Chapitre 1

## Introduction

Le niveau de détail perceptible sur un objet dépend directement de la position relative de l'observateur. Ainsi, la synthèse d'images des scènes très complexes peut être coûteuse si l'on doit modéliser et rendre les objets de manière identique, qu'ils soient proches ou loin de l'observateur. Aussi, ces dernières années beaucoup de travaux ont porté sur la représentation multi-échelle des objets géométriques en particulier dans les environnements interactifs de réalité virtuelle. L'objectif est en général de limiter le coût du processus de rendu sans trop dégrader la qualité de l'image. En effet, pourquoi rendre un arbre composé de 100000 polygones alors que l'observateur, situé à 2 kilomètres, ne perçoit qu'une forme approximative du modèle? Ainsi, le but de la modélisation multi-échelle est de générer des modèles dont la complexité géométrique (et par conséquent le niveau de détail) est paramétrable.

Heckbert [HG94] propose une classification des structures de données utilisables en multi-résolution :

- **les pyramides d'images** : méthode naturelle pour représenter en 2D plusieurs niveaux de détail.
- **les pyramides de volumes** : version 3D des pyramides d'images. Ces structures sont utilisées pour le rendu rapide de données représentées sous forme volumique.
- **textures et modèles de réflexion** : les textures et les modèles de réflexion sont utilisés pour rendre efficacement à la surface des objets des détails qui seraient beaucoup trop coûteux à modéliser explicitement (ce sont les techniques de texture mapping, bump mapping et displacement mapping).
- **images sous plusieurs angles** : un objet est représenté par un ensemble d'images sous différents points de vue permet un rendu rapide multi-échelle mais le nombre de positions étant limité certaines parties des objets peuvent être invisibles ou mal définies.



- **le lancer de rayon** : un objet multi-échelle pour le lancer de rayon est un modèle qui fournit au moteur de rendu le résultat de l'intersection de sa géométrie avec le rayon incident. Le lancer de rayon travaillant dans l'espace image, la complexité serait donc proportionnelle à la surface de l'objet sur l'image. Certaines structures de données permettent un tel traitement, comme les espaces de voxels [Ney96].
- **les modèles polygonaux** : la simplification de modèles polygonaux est le principal axe de recherche en modélisation multi-échelle. De nombreux travaux ont proposé des solutions à ce problème dans le cas de structures spécifiques comme les maillages continus triangulaires. Cependant, la simplification reste un problème difficile dans le cas de topologies arbitraires.

Il convient également de distinguer deux approches dans la modélisation multi-échelle : les modèles pré-calculés et stockés a priori et les modèles interprétés « au vol » lors du rendu de la scène. La première approche a l'inconvénient de nécessiter un volume mémoire important et des problèmes peuvent apparaître lors des transitions entre deux représentations. La seconde approche n'est pas toujours possible, soit il n'existe pas d'algorithme d'interprétation compatible avec la technique de rendu, soit les temps de calcul sont prohibitifs.

En résumé, il n'existe pas aujourd'hui de système multi-échelle général qui résolve les problèmes posés dans un cadre général. C'est un des domaines de recherche de la synthèse d'image encore très actif, en particulier avec la demande très forte d'environnements interactifs pour la réalité virtuelle.

Dans le chapitre suivant, nous définissons la problématique multi-échelle dans le cas des paysages naturels. Nous verrons que ce cadre illustre bien le besoin de techniques capables de s'adapter en fonction d'un niveau de détail. Nous présentons l'une de ces techniques dans le chapitre trois. Il s'agit d'une méthode utilisant un ensemble de textures pour reconstruire rapidement et de façon adaptative un objet complexe.

## Chapitre 2

# Niveaux de détail dans un paysage

Un paysage est un exemple parfait pour illustrer l'intérêt d'une modélisation multi-échelle. Tous les éléments d'un paysage sont complexes, très complexes. De plus, un paysage est composé d'un nombre **très important** d'éléments. Cependant, on remarque immédiatement que dans une vue d'un paysage, un élément est perçu différemment selon qu'il est loin ou proche (cf. figure 2.1). Une approche multi-échelle pour la création de paysage est non seulement possible mais surtout nécessaire.

Nous définissons trois niveaux de détail pour les éléments d'un paysage (principalement la végétation) :

- **le premier plan** : l'observateur peut distinguer des détails sur les éléments de premier plan. C'est le niveau de détail maximum dans une scène de paysage (dans le cas d'un observateur humain standard, pas d'un microscope!).
- **le second plan** : l'observateur peut encore isoler individuellement un élément mais ne distingue plus les détails internes. Il peut identifier l'élément en fonction de critères globaux comme sa taille relative, sa couleur ou sa forme approximative.
- **le plan lointain** : à cette distance, l'observateur ne perçoit plus l'élément mais juste une couverture du relief dont les propriétés de couleur et de texture peuvent rappeler l'élément.

Ainsi, dans une image de paysage, il est possible de rencontrer pour un même élément des niveaux de détail extrêmes. Il serait désastreux pour les performances du système, d'utiliser pour tous les éléments de la scène la méthode de modélisation des éléments de premier plan. Nous avons choisi de faire appel à plusieurs techniques de modélisation pour maintenir un bon niveau de performance sans dégrader de façon significative la qualité de l'image finale. Nous utilisons les techniques suivantes :

- **pour le premier plan**, la méthode à base de systèmes de particules expliquée au chapitre 2 (page 103) est capable de modéliser avec précision les éléments du paysage

que l'observateur voit de près. Ce sont principalement les végétaux, de l'herbe jusqu'à l'arbre.

- **pour le second plan**, nous avons développé une technique basée sur la projection d'une collection de textures. Cette technique expliquée dans la suite de ce chapitre permet de rendre rapidement des objets stockés sous forme d'un ensemble de textures. Elle est particulièrement adaptée au rendu des arbres (cf. chapitre 3, page 137).
- **pour l'arrière plan**, nous utiliserons classiquement des textures. Les textures 2D permettent de rendre très rapidement et pour un coût minime de vastes étendues. Cependant, de nombreux problèmes existent avec cette technique, en particulier la prise en compte de l'illumination ou de l'auto occultation.

Un des problèmes les plus difficiles dans l'utilisation de méthodes différentes pour modéliser un même élément à plusieurs niveaux de détail reste la fusion des représentations à la frontière entre deux plans. En effet, il est important de ne pas voir apparaître de discontinuité au niveau de la transition entre deux représentations d'un même objet. Une solution peu satisfaisante en terme de coût consiste à mélanger les deux représentations par un procédé de composition d'images pour produire ainsi une transition « floue » au niveau de la frontière.

D'autre part, dans le contexte d'une représentation multi-échelle des éléments, le système doit pouvoir décider de la représentation à utiliser. Ce choix dépend bien sûr de la position de l'observateur par rapport à l'objet, des caractéristiques de la projection et des attributs de l'environnement (conditions atmosphériques). Il est également possible de faire intervenir un critère de qualité de rendu dans cette décision.



FIG. 2.1 – Une photographie d'un paysage réel et l'énumération des différentes échelles pour les éléments.



## Chapitre 3

# Les objets multi-textures

Dans ce chapitre, nous allons présenter une méthode pour la représentation et le rendu rapide d'objets, en particulier des éléments naturels comme les arbres. L'information 3D des objets multi-textures est reconstituée à partir d'un ensemble de vues et de cartes de profondeur. Ces cartes contiennent des informations de couleur (rouge, vert et bleu : RVB) ainsi que les données de profondeur ( $Z$ ) pour un point de vue défini. Les données de ces vues peuvent être facilement calculées à partir du rendu de l'objet, mais peuvent aussi être captées par un dispositif matériel dans le cas d'un objet réel.

Deux phases successives et distinctes sont nécessaires pour la mise en oeuvre des objets multi-textures (cf. figure 3.1) :

- **calcul des vues** : l'ensemble des cartes ( $RVB + Z$ ) de l'objet doit être préalablement calculé et stocké. Une méthode de rendu par  $Z$ -buffer pourra fournir les informations de couleur et de profondeur pour un point de vue donné. De plus la matrice de transformation de la projection orthographique associée à chaque vue doit être retenue.
- **rendu rapide de l'objet** : un procédé de re-projection va permettre de rendre l'objet sous un point de vue quelconque à partir d'un nombre restreint de cartes. De la résolution des cartes dépendra le niveau de détail de l'objet reconstruit ainsi que le temps de re-projection. C'est donc une technique de rendu multi-échelle, la précision et la vitesse de rendu pouvant être contrôlées.

### 3.1 Le calcul des cartes

Le calcul de l'image d'un objet et de la carte de profondeur associée ne pose pas de véritable problème et peut aussi bien être réalisé par une technique de  $Z$ -buffer que de lancer de rayons. La question épineuse est la suivante : combien de cartes doivent être calculées et sous quels points de vue pour permettre une bonne reconstruction de l'objet ? Nous essaye-

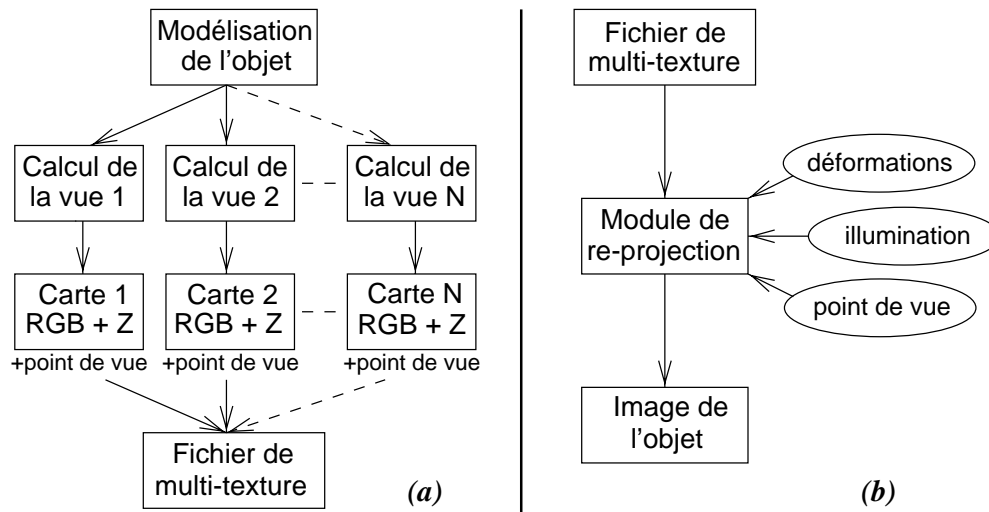


FIG. 3.1 – Mise en oeuvre des objets multi-textures : (a) calcul des cartes, (b) rendu de l'objet dans la scène.

rons d'apporter des éléments de réponse à ce problème dans la section 3.3 (page 142).

Le calcul des cartes va produire pour un objet et pour un point de vue (cf. figure 3.2) :

- l'image de l'objet par projection orthographique;
- la carte des profondeurs de l'objet depuis le point de vue;
- la matrice de transformation utilisée pour la projection orthographique.

Afin de réduire le volume de données, les images pourront être quantisées<sup>1</sup> et les cartes de profondeur compressées. En effet, il est nécessaire de réduire au maximum le volume des données stockées pour un objet multi-texture, le nombre d'objets différents pouvant être important dans une scène complexe (comme un paysage, cf. section 3.4, page 144).

## 3.2 La re-projection des cartes

C'est à partir de l'ensemble des vues pré-calculées que l'on doit reconstruire un objet et le rendre sous un point de vue quelconque. Pour des raisons d'efficacité, il n'est pas question de passer par une forme géométrique 3D intermédiaire de l'objet, mais de générer directement l'image projetée de cet objet. De plus, pour garantir un certain niveau de réalisme, ce rendu doit prendre en compte les conditions d'illumination de la scène. En effet ces objets seront ainsi mieux intégrés dans le contexte de la scène finale.

1. **quantisation** d'une image : réduction du nombre de couleurs composant l'image.

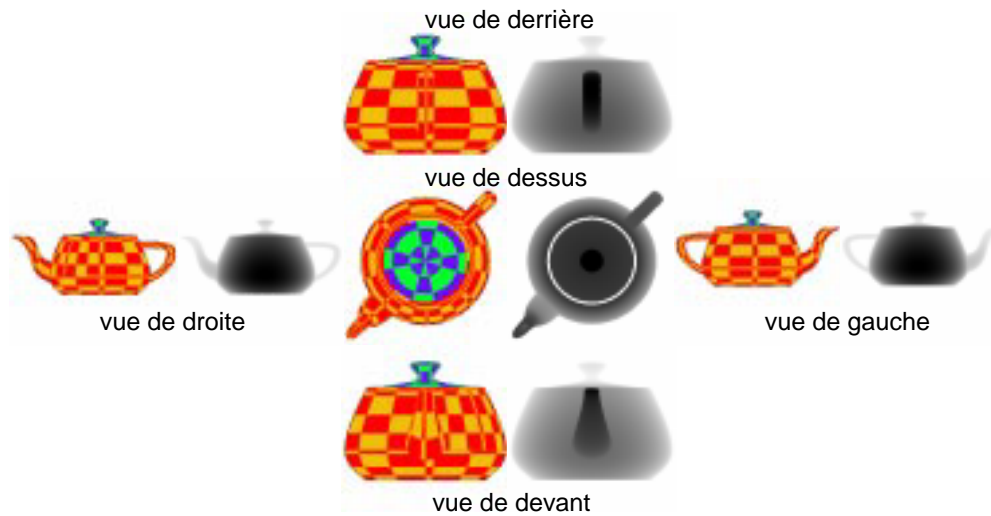


FIG. 3.2 – Cinq vues d'une théière correspondant aux faces d'un cube (hormis la vue de dessous). Le volume total des données pour cet objet multi-texture est d'environ 300 kilo octets (après compression) avec des cartes de résolution  $256 \times 256$ .

Le problème du rendu 3D des objets multi-textures consiste à passer d'un ensemble discret de données  $(u, v, Z[u, v])$  présent dans les cartes de profondeur à une surface dans l'espace scène, puis à projeter cette surface pour former l'image finale. Les transformations successives sont donc :

$$(u, v, Z[u, v])_{texture} \xrightarrow{\text{projection inverse}} (x, y, z)_{scène} \xrightarrow{\text{projection perspective}} (i, j, z)_{image}$$

La projection inverse est simplement l'inverse de la matrice de transformation retenue pour chaque vue. Il est donc possible de composer ces deux transformations successives en une seule opération matricielle. Cette matrice de transformation ( $M_t$ ) de l'espace texture à l'espace image est le résultat du produit des deux matrices de transformation.

Nous détaillons dans les sections suivantes, trois méthodes pour réaliser le rendu d'un objet multi-texture.

### 3.2.1 Construction de micro-polygones

Des micro-triangles vont être formés à partir de trois points adjacents dans les cartes. Ces micro-triangles 3D seront projetés pour former la surface de l'objet dans l'image finale. La couleur aux sommets du micro-triangle sera donnée par l'image RVB de la vue, tandis que les coordonnées projetées seront calculées par  $M_t$  à partir des données  $Z$  de la vue. Cette méthode est intéressante car les micro-triangles peuvent être rendu très rapidement par un algorithme basé sur des tables de pré-calculs [McM95]. Néanmoins, cette technique ne tire



pas vraiment partie de la représentation spécifique des objets multi-textures et s'avère très lourde en calcul (le nombre de micro-triangles est très grand).

### 3.2.2 Tracer de rayon

Les cartes de profondeur sont semblables aux cartes d'élévation que nous avons évoquées dans la section 1.2.2.1 (page 53) sur le rendu des terrains. Nous avons remarqué que de telles structures de données régulières pouvaient être rendues rapidement par une méthode de lancer de rayons [CS93]. En effet, le calcul de l'intersection de la surface et du rayon est réalisé efficacement par un parcours incrémental de la carte 2D. C'est donc une méthode basée dans l'espace image par opposition à la précédente qui opère dans l'espace objet. Cependant, bien que précise, cette approche est assez lente car les rayons doivent parcourir plusieurs vues à la recherche de la première intersection. La complexité dépend donc du nombre de rayons (la surface de l'objet sur l'image finale), de la résolution des cartes et dans une moindre mesure du nombre de cartes.

### 3.2.3 Re-projection directe

La re-projection directe consiste simplement à transformer directement les points de l'espace texture dans l'espace image et à les afficher. En effet, lorsque la taille projetée de l'objet sur l'image finale est inférieure à la résolution des cartes, le nombre de points transformés sera suffisamment important pour former une image consistante. La figure 3.3 montre la re-projection directe d'une seule carte dans le cas où l'objet sur l'image a sensiblement la même taille que la résolution de la carte.

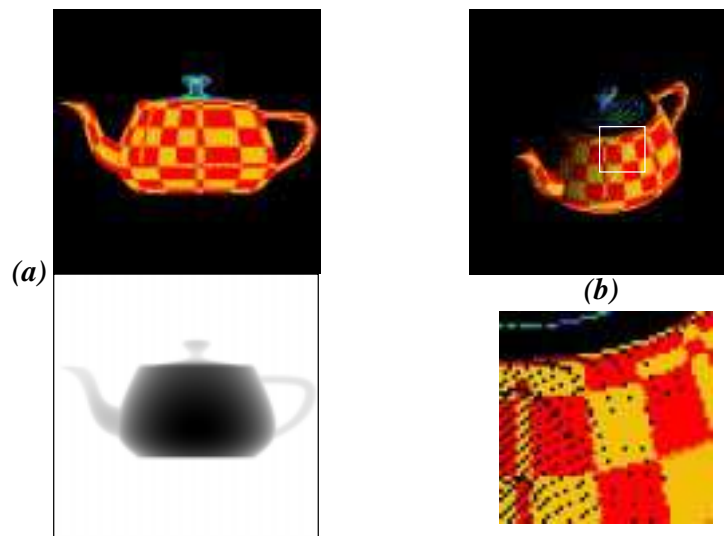


FIG. 3.3 – Image de l'objet multi-texture (b) produit à partir d'une seule carte (RVB + Z) (a). On remarque que les zones qui ne sont pas ou mal renseignées sur la carte apparaissent noires sur l'image finale.

La figure 3.4 illustre la re-projection de 5 vues permettant de former une image de théière. Là aussi, le nombre de points noirs sur la surface dépend directement de la taille projetée de l'objet. On remarque également que des zones qui n'apparaissent sur aucune carte, comme le corps de la théière caché par l'anse, forment des « trous » sur la surface.

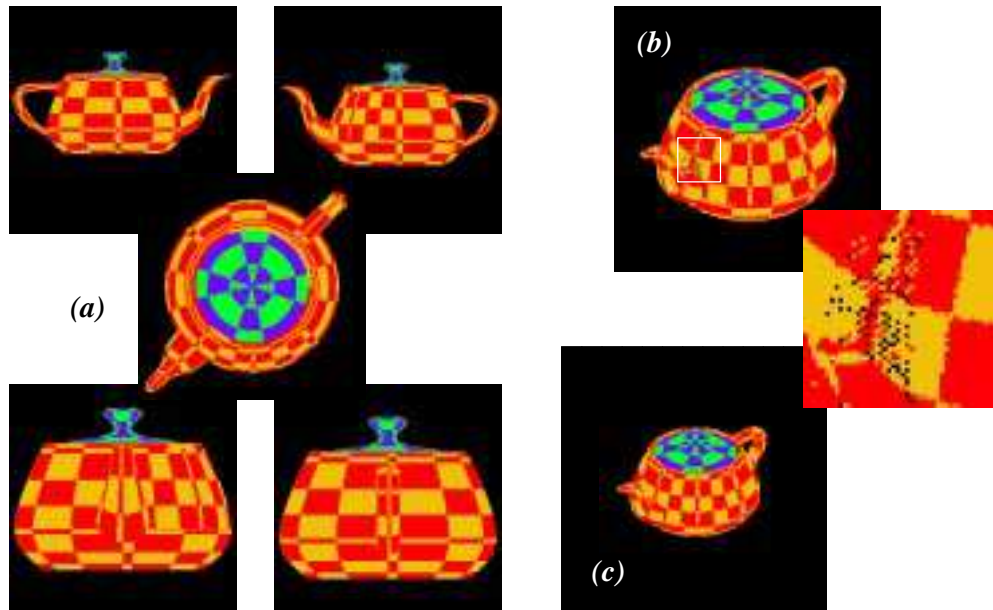


FIG. 3.4 – Images de l'objet multi-texture produit à partir de 5 cartes correspondantes aux faces d'un cube (sauf la vue de dessous) (a). L'image (b) montre le rendu de l'objet, des points non renseignés sont visibles sur le détail. L'image (c) est un rendu du même objet mais avec un autre point de vue : la taille projetée étant plus petite, les points noirs ont disparu (voir aussi planches couleurs page 175).

Cette technique, bien qu'ayant de nombreux inconvénients comme nous allons le voir maintenant, possède un avantage important : la vitesse de rendu. En effet, une multiplication d'un point de la texture par la matrice de transformation combinée suffit pour trouver les coordonnées de ce point sur l'image (il faut également utiliser une division dans le cas d'une projection perspective). Cependant, une optimisation importante est réalisée : les cartes étant parcourues dans l'ordre des lignes et des colonnes, un calcul incrémental du point sur l'image est possible. L'algorithme est donc le suivant :

Comme l'illustre cet algorithme seulement 3 multiplications sont nécessaires pour calculer la re-projection d'un point des textures. Il faut bien sûr rajouter à cela le coût de la transformation perspective (1 division et 2 multiplications) si nécessaire. Nous obtenons des temps de calcul assez faibles, inférieurs à 0.5 seconde, pour le rendu du modèle de théière avec 5 vues (avec élimination des points cachés par Z-buffer).

Les contraintes imposées par cette technique sont moins pénalisantes si on l'utilise dans un contexte de modélisation multi-échelle. On appliquera une telle méthode dans le cas où

---

**Algorithme 3.1** Algorithme de re-projection d'un objet multi-texture.
 

---

```

Soit Mi la matrice de transformation de l'espace scène à l'espace image
Pour chaque vue composant l'objet multi-texture Faire
  Soit T[i,j] la carte des couleurs
  Soit Z[i,j] la carte des profondeurs
  Soit Nx et Ny la résolution de la carte
  Soit Mo la matrice de transformation associée à T et Z
  Mt = Inverse(Mo) * Mi
  Pour v de 0 à Nx-1 Faire
    P = (0,v,Z[0,v]) * Mt
    Afficher( P , T[0,v] )
    Pour u de 1 à Ny-1 Faire
      Soit DeltaZ = Z[u,v] - Z[u-1,v]
      P.x = P.x + Mt[1,1] + Mt[1,3] * DeltaZ
      P.y = P.y + Mt[2,1] + Mt[2,3] * DeltaZ
      P.z = P.z + Mt[3,1] + Mt[3,3] * DeltaZ
      Afficher( P , T[u,v] )
    FinPour
  FinPour
FinPour

```

---

des objets de second ou d'arrière plan doivent être rendu très rapidement (comme les arbres d'une forêt par exemple). Dans ce cas, le problème des « points noirs » évoqué précédemment, sera beaucoup moins perceptible.

Par rapport aux objectifs de notre système, la re-projection directe est la technique la plus adaptée pour le rendu des objets multi-textures.

### 3.3 Problème, amélioration et extensions

Nous présentons maintenant certaines améliorations et extensions pour la technique de re-projection directe. Le problème principal de cette approche reste la présence à la surface de l'objet de « points noirs » qui correspondent à des zones peu ou pas représentées sur les vues. Il est possible d'améliorer significativement la qualité du rendu en générant une interpolation dans l'espace image entre deux points des textures vérifiant un critère de continuité. Ce critère s'appuie sur le fait que deux points voisins font probablement partie d'une même surface et doivent donc être connectés lors du rendu dans l'espace image. Cette information de continuité peut soit être incorporée aux données des cartes, soit être évaluée lors du rendu. Pour réaliser cette évaluation, nous examinons la continuité de la tangente locale (en fait, les variations de la variable DeltaZ, cf. algorithme 3.1).

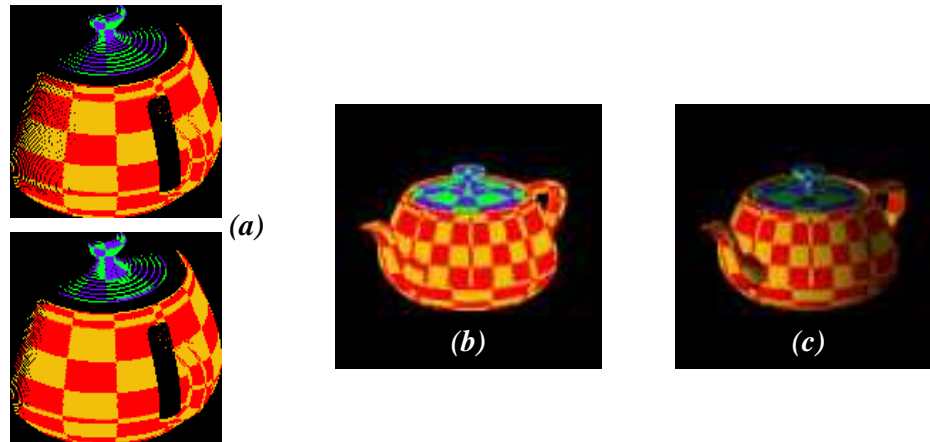


FIG. 3.5 – L'interpolation dans l'espace image des points projetés permet d'améliorer la qualité du rendu, (a) montre le rendu d'une seule vue sans et avec interpolation. Le nombre de points noirs a fortement diminué mais les zones qui n'apparaissent pas sur la carte sont toujours absentes (comme la partie du corps de la théière cachée par l'anse). L'image (b) montre la théière reconstituée par 5 vues interpolées. Enfin, l'image (c) montre l'illumination de l'objet multi-texture (voir aussi planches couleurs page 175).

Comme nous l'avons mentionné en introduction, le rendu de l'objet multi-texture doit prendre en compte les conditions d'éclairage. Cependant, nous ne disposons pas de la normale en chaque point de la surface pour effectuer un calcul d'illumination. En guise d'approximation, nous utiliserons le gradient local évalué avec 4 points adjacents ( $Z[u-1,v]$ ,  $Z[u,v+1]$ ,  $Z[u+1,v]$  et  $Z[u,v-1]$ ) pour déterminer une normale approchée. Le calcul de la couleur finale sera ensuite réalisé par un modèle classique d'illumination (cf. figure 3.5).

Un problème persistant et inhérent à cette technique reste le choix du nombre et des caractéristiques des différentes vues nécessaires à la représentation d'un objet. Trois critères essentiels guident le choix des vues :

- **le type de l'objet** : un objet convexe ou quasi convexe pourra être assez bien décrit par un ensemble de vues. Par contre, certains modèles aux formes complexes échapperont à ce type de représentation, de nombreuses zones de la surface ne pouvant être capturées sans recourir à un nombre prohibitif de vues.
- **la qualité du rendu** : la qualité de la re-projection dépend de la « couverture » de la surface de l'objet par les vues et de la résolution de ces vues.
- **la vitesse du rendu** : les calculs nécessaires à la re-projection sont directement proportionnels au nombre et à la résolution des cartes.

Le choix le plus simple et en général suffisant pour des objets quasi convexes consiste à prendre les vues correspondantes aux projections sur les faces d'un cube englobant (cf. figure 3.4). Ce n'est malheureusement pas suffisant dans le cas d'objets plus complexes, comme l'illustre la figure 3.6. Des vues plus petites représentant une zone ou des détails indispensables doivent dans ce cas être ajoutées à la collection. La méthode n'imposant aucune contrainte quand à la direction ou à la résolution des cartes, il est possible de cadrer au mieux la région que l'on désire ajouter.



FIG. 3.6 – Un objet multi-texture complexe : (a) les 5 vues pré-calculées, (b) deux images de l'objet multi-texture. On remarque des zones non renseignées sur le dossier des sièges. Le rendu de cet objet est réalisé en moins d'une seconde sur station SUN Ultra Sparc. Le modèle polygonal original est composé de 34000 polygones.

### 3.4 Application aux éléments naturels

Les objets multi-textures s'intègrent parfaitement dans notre système de rendu de paysages. En effet, le rendu par les multi-textures permet de créer rapidement des zones de second plan composées d'un nombre important d'éléments. Les objets multi-textures sont utilisés pour rendre à moindre coût la végétation pour des niveaux de détail intermédiaires. Ces objets ne peuvent pas remplacer une modélisation précise et complète nécessaire au premier plan, mais ils sont suffisamment détaillés pour créer une forêt.

La rapidité de rendu d'un objet multi-texture est, dans le cadre des paysages naturels, un avantage important. En effet, il n'est pas envisageable d'utiliser la technique des systèmes de particules et des automates pour rendre les milliers d'arbres présents dans une forêt.

Le problème majeur qui apparaît dans ce contexte concerne le pré-calcul des objets multi-textures. Bien que tous les arbres d'une forêt soit différents, il n'est pas possible de stocker préalablement au rendu de la scène, un nombre équivalent d'objets multi-textures. Il faudra

donc composer la scène avec un ensemble restreint de ces objets. Rappelons toutefois que leur utilisation est limitée aux éléments de second plan, qui sont par définition des objets distants sur lesquels on ne perçoit que peu de détail. De plus, nous proposerons dans la suite, une méthode de déformation d'objet multi-texture afin de pouvoir créer des modèles sensiblement différents à partir de cartes identiques.

### 3.4.1 Calcul des vues

Bien qu'un arbre ou une plante soit un modèle géométrique très complexe (et pas du tout convexe), un nombre restreint de vues est suffisant pour exploiter un objet multi-texture. En effet, ce que l'on cherche à capturer du végétal pour ensuite le rendre rapidement, est plus la forme et la texture de l'objet, que des détails géométriques qui sont de toutes façons imperceptibles à une certaine distance. Ainsi, nos expérimentations ont montré que 5 cartes RVBZ correspondant aux vues de face, de derrière, de droite, de gauche et de dessus sont suffisantes pour créer une image convainquante de l'arbre multi-texture (cf. figure 3.7).



FIG. 3.7 – Cinq vues d'un arbre. Pour chaque vue, une carte de profondeur et une carte des couleurs sont calculées.

Bien sûr, de nombreux détails de l'arbre ne figurent pas sur ces cartes, mais l'apparence extérieure est retenue. Une vue de dessous n'est pas nécessaire, l'observateur étant en général au dessus du sol!

### 3.4.2 Le rendu de l'objet naturel

Nous avons détaillé précédemment la méthode de rendu des objets multi-textures, nous ne reviendrons pas dessus. Toutefois, un objet naturel comme un arbre n'a que peu de points communs avec une théière. Ainsi, la technique d'interpolation n'est pas applicable car il n'y a pas de continuité de surface sur les vues d'un arbre (sauf peut-être localement pour le tronc).

Pour résoudre le problème de l'illumination de l'objet, nous utiliserons le même principe que pour les systèmes de particules : une fonction d'illumination probabiliste (cf. section 2.5.1, page 113). C'est à dire que nous évaluons l'illumination d'un point par sa position par rapport au volume englobant de l'objet (cf. figure 3.8).

Pour varier l'apparence de l'objet, sans avoir à recalculer les nouvelles cartes d'un autre modèle, nous utilisons une fonction continue de déformation de l'espace. Cette fonction déforme localement l'espace 3D changeant ainsi la forme de l'objet tout en gardant la cohérence du modèle. Les fonctions utilisées sont du type  $p' = p + \sin(p_o + p * p_f) * p_a$  avec  $p$  la coordonné du point dans l'espace scène,  $p'$  le point après déformation et  $p_o, p_f, p_a$  les paramètres de la déformation. Le figure 3.8 montre trois objets différents créés à partir de la même définition multi-texture, mais en appliquant au moment de la re-projection une fonction de déformation sur les points 3D.

L'algorithme de re-projection s'adapte bien à de telles extensions et le surcoût induit par ces techniques est négligeable.

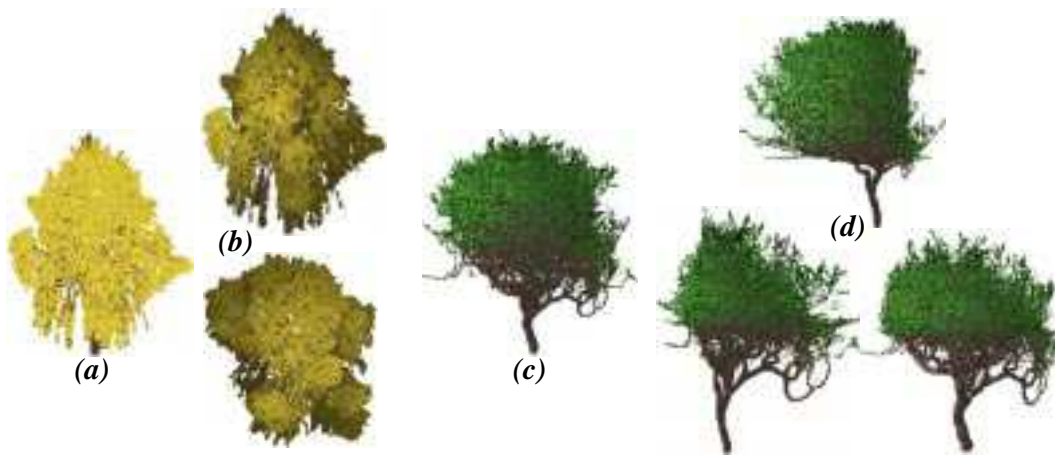


FIG. 3.8 – *Extensions de la méthode de re-projection : (a) une re-projection à partir de 5 vues sans illumination, (b) le même objet multi-texture avec fonction d'illumination probabiliste, (c) un autre objet multi-texture sans déformation, (d) 3 objets différents générés par déformation de l'espace (voir aussi planches couleurs page 175).*

La représentation à base de textures permet également d'adapter le modèle en fonction de la résolution que l'on souhaite obtenir lors du rendu final. Un échantillonnage des cartes de couleurs et de profondeurs permet la construction d'un modèle multi-résolution semblable aux pyramides d'images évoquées précédemment (cf. figure 3.9). Le choix d'une résolution au moment du rendu dépendra de la taille projetée de l'objet sur l'image finale.

Les résultats du chapitre 4 (page 161) font un usage intensif des objets multi-textures, en particulier pour la visualisation des forêts de second plan. Les performances mentionnées accèdent la validité d'une telle approche pour le rendu de ce type de scène.

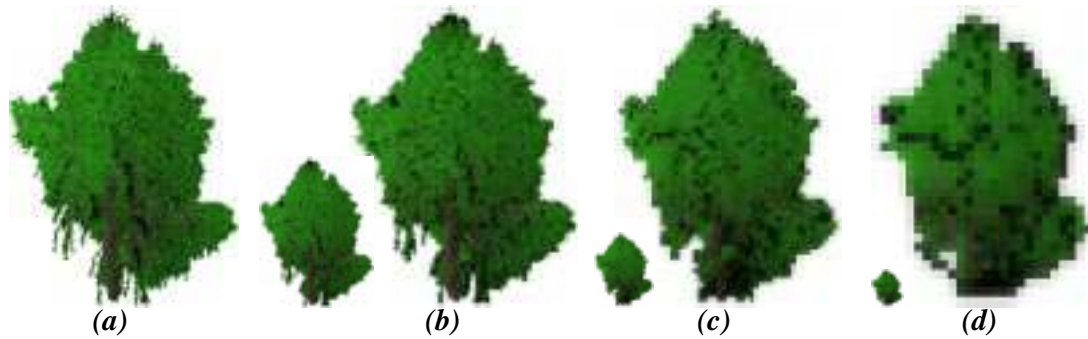


FIG. 3.9 – Rendu multi-échelle des objets multi-textures : (a) rendu original en 0.7 seconde à partir de 5 vues  $256 \times 256$ , (b) le même objet rendu en 0.19 seconde à partir de vues  $128 \times 128$ , (c) rendu en 0.06 seconde pour une résolution de  $64 \times 64$ , (d) 0.01 seconde pour  $32 \times 32$ .

### 3.5 Conclusion sur les objets multi-texture

Nous avons proposé une nouvelle méthode pour stocker et rendre rapidement des objets complexes. Ces objets sont rendus sur l'image finale par le parcours d'un ensemble de cartes, définissant à la fois la couleur (texture) de la surface mais aussi la position dans l'espace. Ces vues sont initialement calculées par des méthodes classiques de rendu et stockées en prévision de leur utilisation future. Cette technique très simple est également très rapide. En contrepartie, elle souffre d'un certain nombre de défauts : la qualité moyenne du rendu sous certaines conditions, la difficulté d'incorporer un calcul précis de l'illumination, le volume important de données à stocker.

Cependant, nous pensons que dans un contexte d'utilisation précis, comme le rendu d'éléments naturels de second plan, cette technique est très intéressante. Elle permet de rendre à moindre coût des portions d'une scène très complexe avec un niveau de réalisme suffisant. En effet, dans des scènes mettant en oeuvre un nombre très important d'éléments, comme c'est le cas des paysages naturels, la précision du détail est moins importante que l'apparence visuelle globale.

L'utilisation des objets multi-textures passe obligatoirement par une étape préalable de construction d'une bibliothèque d'objets. Dans notre système de création de paysages naturels, cette bibliothèque sera construite automatiquement au fur et à mesure des besoins en éléments. Nous disposerons donc à tout moment d'un ensemble d'objets multi-textures pour le rendu rapide d'éléments de second plan, cet ensemble pouvant facilement être étendu.

Parmi les extensions possibles de cette technique, nous envisageons son application au rendu rapide de nuages. Nous étudions également un outil qui permettrait de déterminer automatiquement, pour un objet donné (sous forme polygonale par exemple), le nombre et



les caractéristiques des vues nécessaires pour obtenir une bonne représentation de sa surface.

Une autre utilisation possible des objets multi-textures pourrait être la visualisation interactive d'objets complexes, tâche très lourde lorsqu'ils sont représentés sous forme polygonale. Il paraît possible d'obtenir des animations temps réel sur des ordinateurs ne disposant d'aucune capacité d'accélération graphique.

En 1995, Max [NO95] a publié une méthode pour le rendu d'arbres très proche de celle que nous avons développée. Les raisons qui ont motivé ce travail sont sensiblement les mêmes que les nôtres, à savoir la possibilité de rendre rapidement des objets très complexes, et plus particulièrement des arbres. La technique décrite par Max repose sur le calcul d'un ensemble de vues, qui comprennent pour chaque pixel, la couleur, la profondeur et un codage de la normale en ce point. Le rendu de cet objet est également réalisé par re-projection de tous les points. C'est donc une méthode très semblable à la notre qui a été développée parallèlement. Cependant, les résultats obtenus par Max ne permettent pas l'utilisation quasi temps réel des objets. En effet, pour avoir une bonne qualité de rendu, un nombre important de vues est calculé dans une résolution élevée. Le volume de donnée est donc très important et le temps de rendu également (cf. figure 3.10).

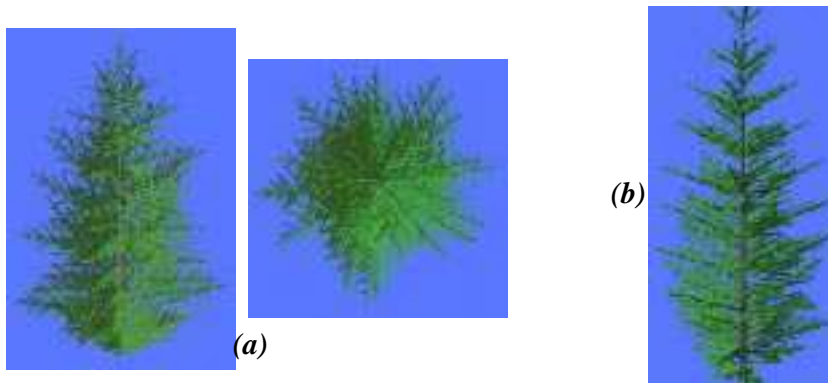


FIG. 3.10 – *Arbre reconstruit par la technique de Max [NO95]: (a) deux vues d'un arbre, (b) l'arbre reconstruit à partir d'un ensemble de vues. Chaque vue à une résolution de  $300 \times 600$  représentant 2500 Ko de données. La reconstruction d'un modèle à partir de 14 vues est effectuée en 6,5 secondes sur station SGI 4D/35 (processeur MIPS R3000).*

## Chapitre 4

# Conclusion

Le développement de techniques multi-échelles est aujourd'hui un sujet de recherche important en infographie. Le besoin de modéliser et de rendre des scènes de plus en plus importantes est bien sûr à l'origine de cette préoccupation. Même si la puissance des ordinateurs augmente significativement, elle reste bien en deçà des performances permettant la création rapide d'images d'environnements très complexes. De plus, il y a une émergence forte aujourd'hui d'un domaine d'application de l'infographie, à savoir la réalité virtuelle. Les systèmes de réalité virtuelle doivent être capables de visualiser en temps réel, c'est à dire avec des fréquences de l'ordre de 10 à 50 images par seconde, des scènes 3D. Il y a donc un intérêt réel dans le développement de techniques permettant d'économiser des calculs en adaptant la qualité du rendu d'un objet en fonction de son importance dans l'image perçue par l'utilisateur.

Notre technique des objets multi-textures constitue peut être une réponse intéressante à ce problème. Elle trouve déjà sa place dans un système (non temps réel) de rendu de paysage, mais ses caractéristiques permettent d'envisager d'autres utilisations.

Au delà des techniques de rendu adaptatives, l'approche multi-échelle peut s'appliquer aux développements de méthodes de représentation et de modélisation. L'objectif dans ce cas est de définir avec la même représentation, la géométrie et les attributs d'un objet à plusieurs échelles. C'est à dire, n'avoir qu'une seule définition pour l'objet depuis le niveau microscopique jusqu'aux échelles les plus grandes, comme par exemple des nervures d'une feuille jusqu'à la forêt amazonienne. Les textures volumiques développées par Neyret [Ney96] semblent une approche prometteuse à ce type de problème.



## Sixième partie

# Un système pour la synthèse d'images de paysages



# Chapitre 1

## Introduction

Nous avons présenté une méthode à base de systèmes de particules pour la modélisation et le rendu d'éléments naturels de premier plan (cf. chapitres 2 et 3, pages 103 et 121). Nous avons également proposé la technique des objets multi-textures pour la visualisation rapide d'éléments de végétation de second plan (cf. chapitre 3, page 137). Il reste à définir un système capable de composer un paysage avec ces techniques et ce, de manière simple et efficace.

Nous avons défini en introduction les contraintes et les fonctionnalités les plus importantes de ce système :

- le système doit traiter rapidement les problèmes triviaux de visibilité. De plus, pour tous les éléments potentiellement visibles, un niveau de détail doit être déterminé et la méthode de modélisation-rendu choisie en conséquence.
- le système doit être capable de distribuer les calculs sur une architecture parallèle.
- il doit être possible d'étendre facilement les fonctionnalités de ce système, en particulier, en définissant de nouveaux éléments.
- dans le cas d'animations par exemple, les temps de calcul pouvant être longs, une interface conviviale doit permettre à l'utilisateur de connaître l'état du système et l'avancement des traitements.

Le système que nous avons construit et qui fonctionne actuellement respecte ce cahier des charges. Nous allons voir dans le chapitre suivant les détails de son architecture. Nous présenterons ensuite les techniques que nous avons utilisées pour sa réalisation, en particulier au niveau des communications et de l'interface avec l'utilisateur. Finalement, nous montrerons quelques résultats illustrant les qualités d'une telle organisation dans le cas du rendu de paysages naturels.



## Chapitre 2

# L'architecture

Le système est chargé de produire des images de synthèse d'un paysage à partir de sa description symbolique et des caractéristiques d'observation. On peut interpréter ce processus comme l'application d'une succession de « filtres » chargés de convertir une représentation donnée d'une scène 3D dans une autre pour finalement aboutir à une image. La figure 2.1 illustre les modules successifs permettant le rendu d'un paysage, depuis sa définition symbolique sous forme cartographique jusqu'à l'image de ce paysage du point de vue de l'observateur.

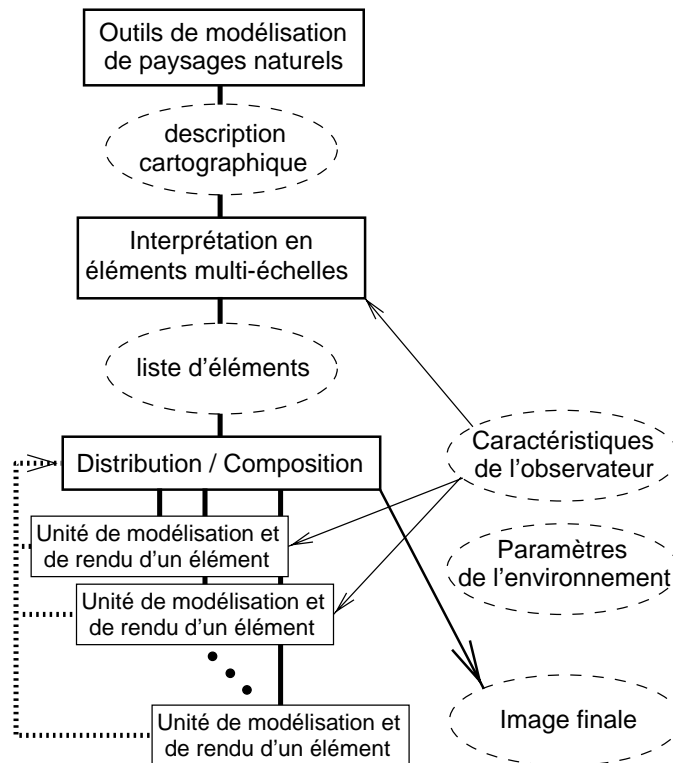


FIG. 2.1 – Les modules du système.



Le premier module, appelé « **Outils de modélisation de paysages** », est chargé de produire une description de la scène. Cette description peut être définie sous forme de cartes représentant différents niveaux de description : la carte des altitudes (modèle numérique de terrain), une carte de couverture du relief (pierres, gazon, eau ...), une carte de végétation et une ou plusieurs cartes d'atmosphère (nuages). Cette structuration est un exemple, nous ne nous sommes pas intéressés dans cette thèse à ce niveau de modélisation. En effet, il existe déjà dans le domaine des systèmes d'information géographique (SIG) des conventions et des logiciels permettant de spécifier de telles descriptions.

C'est au niveau du second module, « **Interprétation en éléments multi-échelles** », que commence réellement l'implémentation que nous avons réalisée. La tâche de ce module est de convertir une représentation symbolique du paysage sous forme cartographique par exemple, vers une représentation en liste d'éléments multi-échelles. Cette interprétation dépend des caractéristiques de l'observateur, en particulier de sa position et de la direction du regard. En effet, ce module est capable d'une part d'éliminer les éléments trivialement invisibles (clipping par la pyramide de vue), et d'autre part, d'évaluer en fonction de la distance à l'observateur, un niveau de détail pour un élément. D'autres optimisations peuvent être envisagées pour améliorer les performances du système : éliminer directement les éléments cachés par le relief ou par des conditions atmosphériques particulières (brouillard par exemple).

Le module de « **Distribution et composition** » commande la modélisation et le rendu des éléments individuels. L'essentiel des coûts dans un système de rendu de paysages est concentré au niveau de la synthèse des éléments. Nous avons donc choisi de distribuer cette tâche sur plusieurs unités de traitement, ce coût étant alors réparti sur l'ensemble d'un réseau d'ordinateurs. Comme nous l'avons expliqué dans la section 4.2.4 (page 90) sur les systèmes de rendu, la parallélisation des traitements est rentable s'il y a adéquation entre la granularité des données échangées et les débits possibles entre les unités du réseau. Dans notre cas, les données échangées définissent un élément, sous forme de paramètres en entrée des unités et sous forme d'image en sortie. L'image d'un élément est ensuite composée dans l'image de la scène (cf. figure 2.2). Le réseau est constitué d'un ensemble hétérogène d'ordinateurs reliés par un réseau classique de type ETHERNET. Les temps moyens de modélisation et de rendu d'un élément étant de l'ordre d'une dizaine de secondes et les transferts de données de l'ordre de 100 Ko, ce type de réseau est bien adapté à ce volume de communication.

Les unités de « **Modélisation et de rendu** » effectuent les calculs nécessaires à la création d'une image de l'élément. Ces modules utilisent les techniques des systèmes de particules (cf. partie IV, page 101), des objets multi-textures (cf. chapitre 3, page 137) ou des textures pour produire une image de l'élément. Le choix de la technique est imposé par le module d'interprétation qui a déterminé, pour tous les éléments devant être modélisés

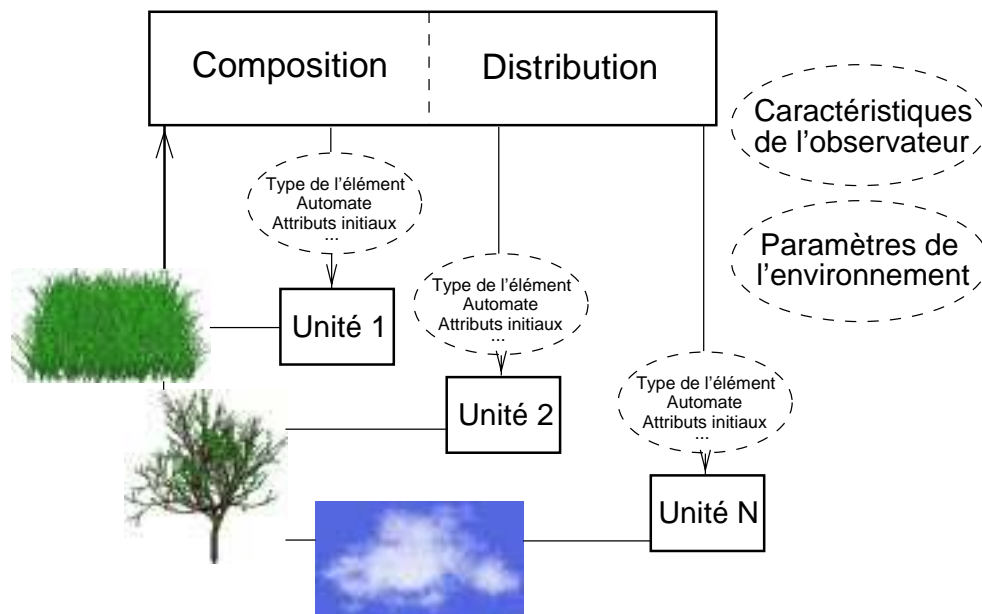


FIG. 2.2 – *Le module de distribution/composition fait parvenir aux unités de traitement les données définissant un élément. En retour, ces unités envoient l'image calculée de l'élément. La modélisation et le rendu tiennent compte des caractéristiques de l'observateur et des paramètres de l'environnement.*

et rendus, un niveau de détail. Les caractéristiques de l'observateur définissent la projection de l'espace 3D dans l'espace de l'image. Les paramètres de l'environnement sont les attributs courants et généraux du paysage, comme le temps global, la présence de vent ou de brouillard, les caractéristiques de l'éclairage ... Ces attributs affectent directement le processus de modélisation et de rendu et sont pris en compte si la technique de création de l'élément le permet.



## Chapitre 3

# Mise en œuvre

Notre système est composé de deux applications indépendantes qui fonctionnent selon le principe client-serveur : le manager et l'unité de traitement. Le manager est le cœur du système et regroupe les modules d'interprétation et de distribution/composition dont nous avons parlé précédemment. L'unité de traitement est la partie du système qui est distribuée sur un réseau d'ordinateurs. Elle effectue la modélisation et le rendu d'éléments naturels multi-échelles. Ces deux applications dialoguent par l'intermédiaire d'un protocole bi-directionnel simple, le manager envoyant des ordres aux unités qui lui renvoient en retour le résultat des calculs (cf. figure 3.1).

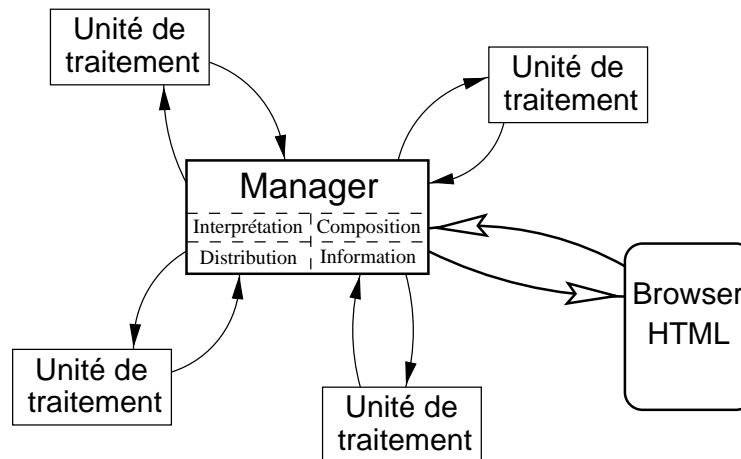


FIG. 3.1 – Les deux applications du système : le manager qui contrôle et commande, l'unité de traitement qui calcule. De plus, un browser HTML est capable de renseigner l'utilisateur sur l'état du système.

Le manager est également capable de fournir des informations sur l'état d'avancement des travaux par l'intermédiaire d'un service Web. Un simple browser de page Web permet de consulter localement ou à distance les informations courantes sur l'état du système.

La réalisation de ce système a nécessité assez peu de programmation. La librairie implé-

mentant les fonctions géométriques de bas niveau ainsi que le moteur de rendu est composée de 4000 lignes de langage C standard. Le code source du manager représente 1100 lignes de C et celui de l'unité de traitement 2500 lignes. Aucune librairie externe n'est utilisée à part les ressources systèmes standards. La partie communication a été réalisée à bas niveau à l'aide des sockets pour permettre un portage facile sur toute station UNIX. Ces applications ont été compilées avec succès et sans changement sur une vaste gamme de plateformes :

- stations SUN Sparc et Ultra Sparc sous SunOS et Solaris
- stations SUN Sparc sous NextStep
- stations Intel Pentium sous NextStep
- stations NextColor sous NextStep
- stations Dec Alpha sous OSF

La programmation a été effectuée en privilégiant deux aspects : la performance et l'extensibilité. En effet, il est important que le système et en particulier l'unité de traitement calcule aussi rapidement que possible afin de pouvoir rendre des paysages complets comportant un grand nombre d'éléments. Par ailleurs, comme nous l'avons remarqué en introduction de ce document, il n'est pas possible dans le cadre d'une thèse de réaliser un système qui prenne en compte tous les éléments d'un paysage naturel. Il est donc pratique de pouvoir facilement étendre le système en ajoutant des nouvelles méthodes de modélisation d'éléments ou des nouvelles fonctionnalités au moteur de rendu. Notre programmation très modulaire des applications permet simplement d'étendre les capacités du système.

## Chapitre 4

# Résultats

De nombreux tests ont été réalisés afin d'éprouver l'efficacité et la souplesse du système. Toutes les images présentées dans ce document ont été calculées par ce système et n'ont nécessité aucune retouche a posteriori. Un tel système de modélisation et de rendu distribué prend tout son intérêt quand des scènes très importantes doivent être traitées. A titre d'exemple, nous avons défini un paysage représentant une zone d'environ  $100 \text{ km}^2$ . En plus d'un modèle numérique de terrain précis (cf. figure 4.1), des cartes de couverture du sol définissent la présence, la densité et le type de végétation. D'autres cartes permettent de spécifier l'apparence du ciel en donnant la distribution et le type de nuage pour une zone. Il reste bien sûr possible d'imposer des éléments individuels spécifiques à des emplacements particuliers. Les planches couleurs de la page 176 présentent quelques images issues d'une séquence animée.

### La représentation et le rendu du terrain

Le volume de données représentant le modèle du terrain étant très important, une approche multi-résolution a permis de limiter le coût du rendu. L'ensemble du terrain est divisé en zones plus petites mais de résolution plus élevée. Le choix d'un niveau de précision est réalisé par le module d'interprétation en fonction des paramètres de l'observateur. Des détails encore plus fins sont obtenus par un procédé d'amplification fractale utilisant localement l'algorithme de Miller (cf. section 1.2.1, page 47).

### Le calcul des ombres portées

Le rendu des ombres portées par la technique du Z-buffer nécessite le pré-calcul de cartes de profondeur (cf. section 2.1.1, page 66). Cependant, pour une scène de la taille d'un paysage, il s'avère impossible de disposer en tout point d'une donnée précise sur l'éclairage (la carte d'ombre nécessaire serait énorme). Nous avons choisi de pré-calculer, en plus d'une carte d'ombre globale de taille raisonnable, plusieurs autres cartes montrant

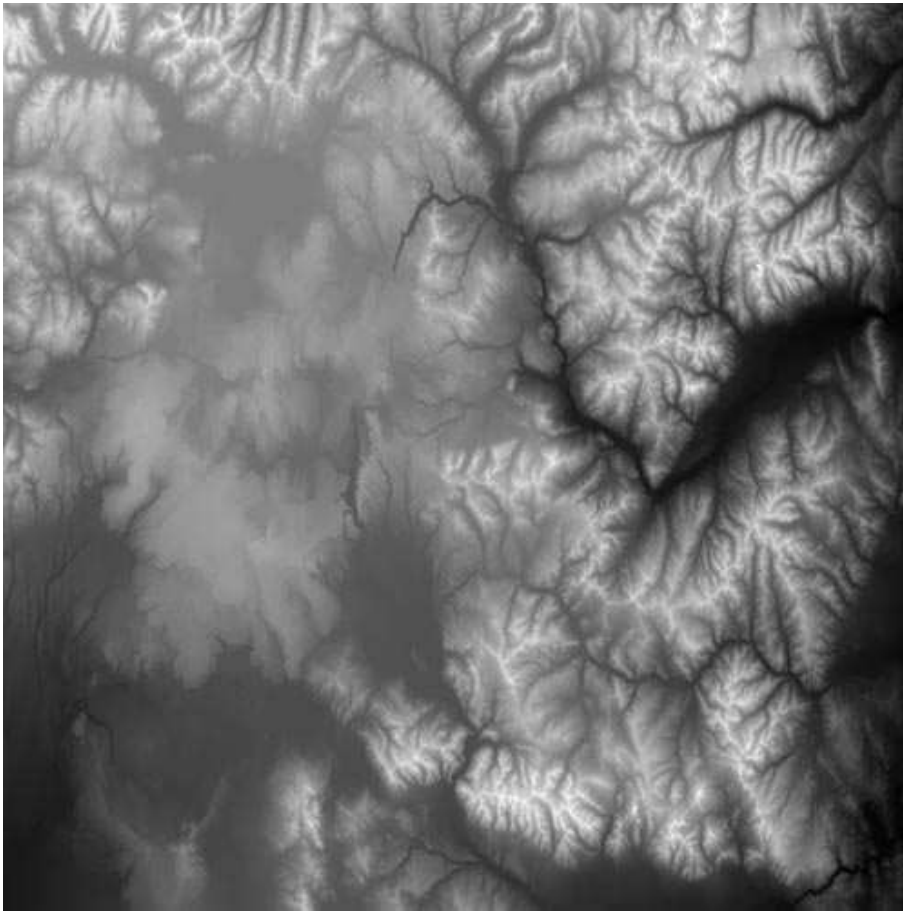


FIG. 4.1 – *Modèle numérique de terrain. Cette zone représente les altitudes d'une partie du parc de Yellowstone aux États-Unis. Il mesure sur une grille de 5 mètres de résolution une surface d'environ 100 km<sup>2</sup> (voir aussi planches couleurs page 176).*

des zones particulières pour lesquelles nous désirons un rendu précis des ombres portées. Au moment du rendu, le système sélectionnera la carte la plus adaptée, c'est à dire celle offrant la meilleure définition au point demandé.

## Performances

Malgré des techniques efficaces et des calculs en parallèle, les temps de rendu pour des scènes complètes peuvent être importants. Pour des images comme celles présentées page 176, 15 minutes de calcul sont nécessaire sur un réseau de 7 stations de classe SUN Sparc 10. Les éléments les plus coûteux à synthétiser sont les étendues d'herbes et les nuages (par systèmes de particules). Viennent ensuite les éléments de végétation (par systèmes de particules) et les reliefs (par maillage de polygones). Les objets multi-textures sont les éléments les moins coûteux du processus.

Pour des images hautes résolutions, la bande passante du réseaux devient également un facteur important de performance. Ainsi, au delà d'un certain nombre d'unités de calcul, l'utilisation de réseaux rapides style ATM serait intéressante.





Septième partie

**Conclusion**



Le but de cette thèse était dans un premier temps de tenter une réflexion sur le problème de la synthèse d'images réalistes de paysages naturels. Cette réflexion était nécessaire car dans le domaine de l'infographie ce sujet n'a pas encore été traité dans son ensemble. Comme nous l'avons vu dans ce rapport, les composants d'un tel système sont nécessairement interdépendants, les besoins des uns étant des contraintes pour les autres. L'avantage d'une approche globale du problème permet de proposer une solution simple, efficace et ouverte.

La première étape de cette réflexion a été la recherche des spécificités du problème de rendu de paysages. C'est-à-dire, chercher les facteurs qui rendent si difficile la production d'images de synthèse réalistes de paysages naturels. Nous avons vu que les obstacles étaient multiples : complexité à toutes les échelles, aussi bien spatiale que temporelle, grande diversité des éléments naturels, difficulté pour décrire de façon uniforme et simple ces éléments... Ces problèmes en tête, nous avons défini ce que devait être un système de rendu de paysages. Nous avons alors pu dégager un certain nombre de principes et de contraintes qui ont guidé nos recherches : la modélisation doit s'appuyer sur des paramètres concrets de la morphologie des objets, elle doit pouvoir intégrer des influences externes et le rendu doit être direct. De plus, ce rendu doit être réaliste et le processus complet de création d'un élément doit être rapide. Parallèlement, nous avons fait un état de l'art complet et détaillé des méthodes et des techniques qui proposaient de modéliser et/ou de rendre des éléments naturels. La conjonction de ces deux analyses nous a permis de conclure qu'il n'existait pas aujourd'hui de méthode parfaitement adaptée aux contraintes que nous avons fixées. Certaines de ces techniques donnent de très bons résultats dans des domaines plus ou moins étendus de la synthèse des éléments naturels, mais aucune n'avait les caractéristiques idéales pour s'intégrer dans un système dédié au rendu de paysage.

Après cette constatation, la deuxième phase de ce travail a été de proposer une approche qui réponde aux contraintes fixées. Notre choix s'est alors porté sur une utilisation

des systèmes de particules. En effet, les systèmes de particules ont une caractéristique intéressante : la possibilité de générer des objets ou des effets très complexes à partir de principes simples. Notre idée est la suivante : les trajectoires de particules ne peuvent-elles pas modéliser des éléments naturels complexes et en particulier les végétaux ? C'est possible, mais à condition de contraindre fortement ces trajectoires et l'évolution des attributs des particules pour aboutir à l'objet voulu. En effet, bien que très divers, des éléments naturels comme les végétaux héritent par leurs gènes d'une structuration qui contrôle leur forme.

Nous nous sommes alors demandé par quel formalisme spécifier le comportement des particules et par conséquent, l'évolution du système de particules. En examinant le problème, on peut s'apercevoir que la difficulté est moins de définir un comportement pour une particule que les changements de comportement. En effet, le principe des systèmes de particules impose que le comportement d'une particule (c'est à dire la fonction d'évolution de ses attributs) soit simple. Il est facile, par une trajectoire simple, de dessiner la forme d'une branche. Ou même, par une seule particule, de représenter une feuille. Le problème est donc bien au niveau des changements de comportement, de la création de nouvelles particules ou de la mort de particules. C'est à partir de ce constat que nous avons pensé au formalisme des automates d'états pour spécifier le comportement du système. Le principe est simple : chaque particule à un instant donné est attachée à un état représentant un comportement, les transitions entre les états sont des changements de comportement avec éventuellement la création de particules.

Cette structure définit totalement la vie du système, depuis la naissance des particules jusqu'à leur mort en passant par un ou plusieurs comportements.

Par les automates, nous avons atteint notre objectif : structurer les systèmes de particules. Dans un automate, les états et les transitions sont en fait des traitements (appelés actions dans le cas des états et fonctions pour les transitions) qui vont faire évoluer les attributs des particules et donner naissance à de nouvelles particules.

Nous avons appliqué ce formalisme à la modélisation des végétaux. Nous nous sommes alors aperçu en écrivant les actions et fonctions, que la topologie de l'automate reflétait la morphologie de la plante. C'est à dire qu'un état définit un niveau de ramification et les transitions la création d'un branchement entre deux niveaux de ramification. Il est ainsi possible de décrire facilement l'automate en observant la morphologie d'un végétal. Les attributs des particules ainsi que les définitions des actions et fonctions sont essentiellement des critères géométriques ou définissant une transformation géométrique. On retrouve ainsi des angles de branchement, des rapports de taille, des probabilités comme dans beaucoup de méthodes de modélisation de structures ramifiées.

Les particules que nous utilisons étant toujours indépendantes et les évolutions toujours fonction de leurs attributs, les contraintes liées au rendu (rendu direct sans stockage de

géométrie) et à la modélisation (intégration d'influences spatiales externes) sont respectées. C'est le grand avantage de cette méthode : elle garde les attraits des systèmes de particules (indépendance, comportement simple, rapidité) tout en permettant la modélisation d'objets complexes structurés et contraints.

Nous interprétons graphiquement une particule par une sphère qui reflète ses attributs (position, taille, couleur, type ...). Le rendu spécialisé que nous avons développé est donc très optimisé pour cette primitive. De plus, le processus de modélisation étant peu coûteux en calcul, le temps nécessaire à la synthèse d'un végétal est faible bien que le modèle créé soit complexe et le rendu de qualité. Cette possibilité de rendre rapidement un grand nombre de particules nous a incité à adapter le couple « système de particules - automate » pour la synthèse de nuages. Comme les végétaux, ce sont des éléments importants d'un paysage. Dans le cas de nuages, les particules sont représentées par des sphères semi-transparentes dont l'accumulation contrôlée par l'automate va créer la forme et l'aspect du nuage.

Que ce soit au niveau de la modélisation ou du rendu, nous faisons un usage intensif de fonctions aléatoires. Le rôle de ces fonctions est d'introduire la diversité dans l'éventail des objets produits. Dans notre système, comme dans la nature, la forme d'un élément ne dépend que de deux choses : la graine et l'environnement. Des conditions strictement identiques (ce qui n'existe pas naturellement) produiront un résultat strictement identique. Cette reproductibilité nous est bien sûr indispensable pour l'animation.

Bien que raisonnablement rapide, la méthode à base de systèmes de particules pour la modélisation et le rendu ne doit pas être utilisée pour rendre tous les éléments d'un paysage. En effet, parmi l'ensemble des éléments que nous percevons d'un paysage, peu ont besoin d'être modélisés et rendus au niveau de détail maximum. Pour résoudre ce problème, nous avons proposé de stocker, puis de visualiser les éléments sous forme d'objets multi-textures. Un objet multi-texture est composé d'un ensemble de vues qui stockent non seulement la couleur mais aussi la profondeur de chaque pixel. Une méthode de re-projection rapide permet de rendre efficacement l'objet sous un point de vue quelconque tout en intégrant les caractéristiques courantes de l'illumination. Cette technique nous permet d'intégrer rapidement un grand nombre d'éléments de second plan dans une scène avec une qualité de rendu suffisante.

Nous avons remarqué initialement que la viabilité d'un système de rendu de paysages naturels passait obligatoirement par le développement d'un logiciel spécialisé. Le système que nous avons réalisé permet d'interpréter une description de paysage en élément multi-échelles modélisés et rendus sur des unités de traitement distribuées sur un réseau d'ordinateurs. L'architecture mise en œuvre peut orchestrer le rendu de scènes très importantes comportant un nombre élevé d'éléments.

En résumé, nous avons proposé un ensemble de solutions adaptées au problème de la synthèse réaliste de paysages naturels. Nous couvrons la modélisation et le rendu d'éléments naturels par systèmes de particules, et le rendu rapide d'objet par une technique multi-texture. Enfin, un système distribué intègre ces méthodes pour produire efficacement des images et des animations de paysages naturels.

A ce niveau de réalisation, il est clair que de nombreuses extensions restent à imaginer avant de prétendre couvrir la diversité naturelle d'un paysage. Nous envisageons l'adaptation des systèmes de particules contrôlés par automates à la modélisation et au rendu d'autres phénomènes naturels comme les rivières et cascades. D'autre part, des recherches doivent être entreprises sur le problème de la représentation multi-échelle des éléments naturels, en particulier pour les zones d'arrière plan où les techniques à base de plaquage de textures ne sont pas satisfaisantes.

Bien que nos modèles soient résolument empiriques, notre approche permet de créer des images de paysages avec un niveau de réalisme et de complexité qui échappe encore aux autres méthodes dédiées à la synthèse d'éléments naturels. C'est principalement le problème du coût de ces méthodes qui impose le choix de techniques empiriques ou approchées, et il est probable que cet obstacle reste encore une contrainte forte pour les années à venir.

Huitième partie

**Planches couleurs**





## Planches couleurs 1 : état de l'art



Fig. 1.8, page 25 : moteur de croissance botanique [dREF+88].



Fig. 1.14, page 30 : L-systems [PLH88].



Fig. 1.19, page 35 : modèle géométrique [WP95].



Fig. 1.28, page 43 : systèmes de particules [RB85].

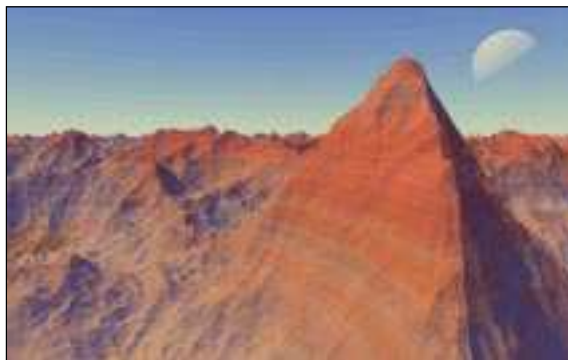


Fig. 1.35, page 52 : fonctions stochastiques [MKM89].



Fig. 1.42, page 59 : simulation de nuages [NYE96].

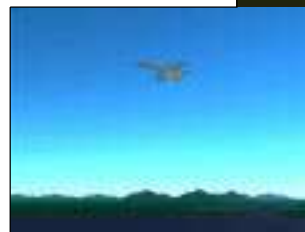
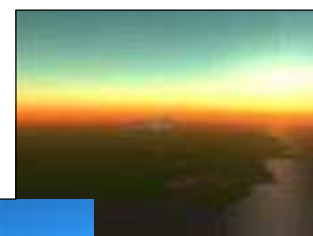


Fig. 1.45, page 61 : simulation de ciels [NDKY96].



Planches couleurs 2 : systèmes de particules pour la synthèse d'éléments naturels



Fig. 2.8, page 112 : modélisation avec prise en compte d'obstacle.



Fig. 2.12a, page 114 : illumination probabiliste.



Fig. 2.10, page 113 : influence du vent.

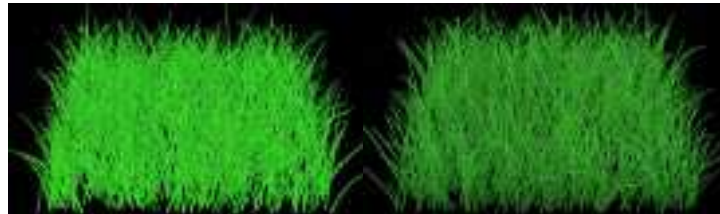


Fig 2.12.b, page 114 : illumination probabiliste.



Fig. 2.14 et 2.15, page 116 : exemples.

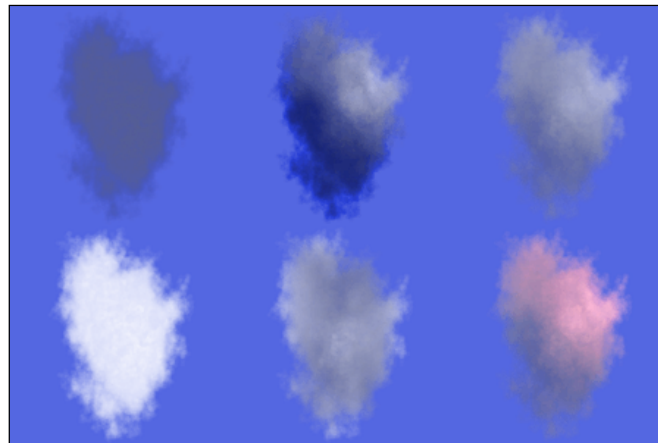


Fig. 3.4, page 124 : rendu de nuages avec illumination.

## Planches couleurs 3 : objets multi-textures

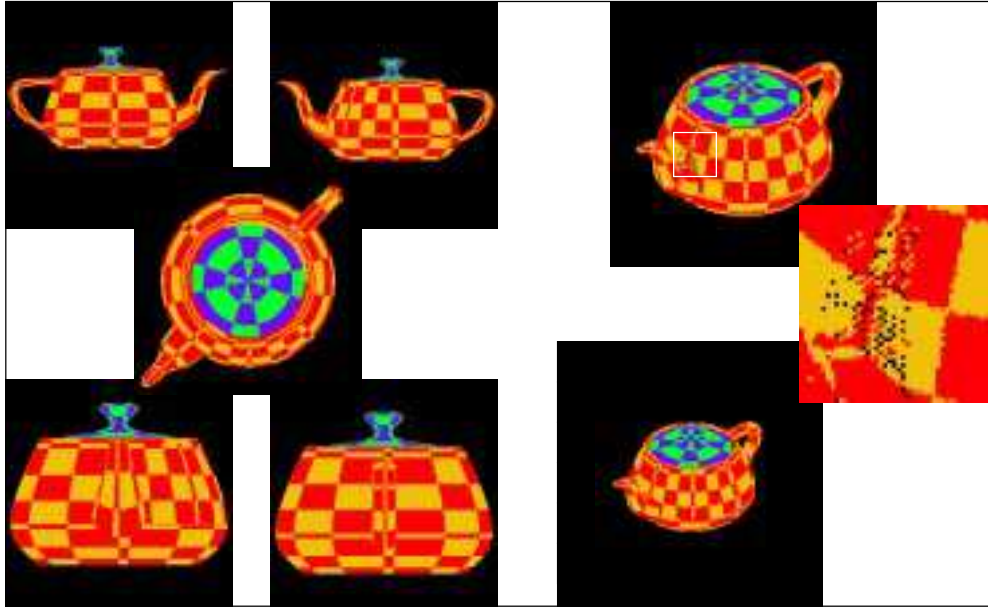


Fig. 3.4, page 141 : reconstruction d'un objet multi-texture à partir de 5 vues.



Fig. 3.5, page 143 : interpolation et illumination.

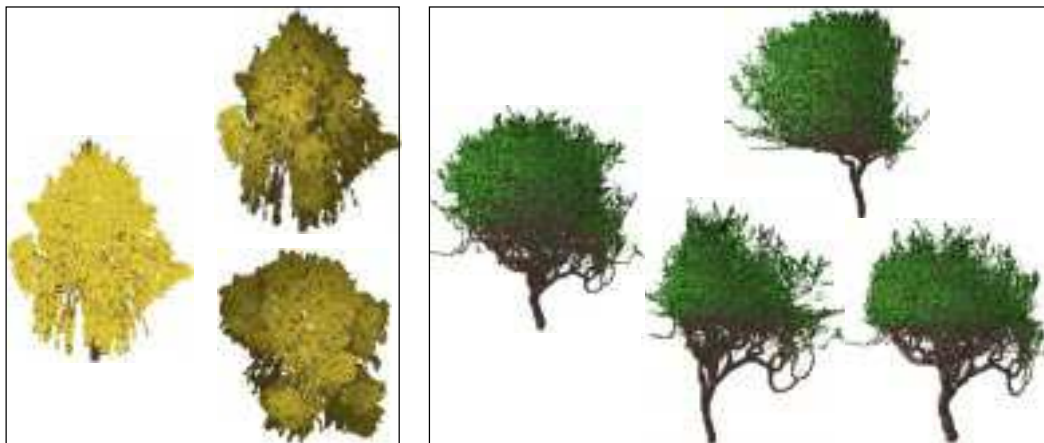
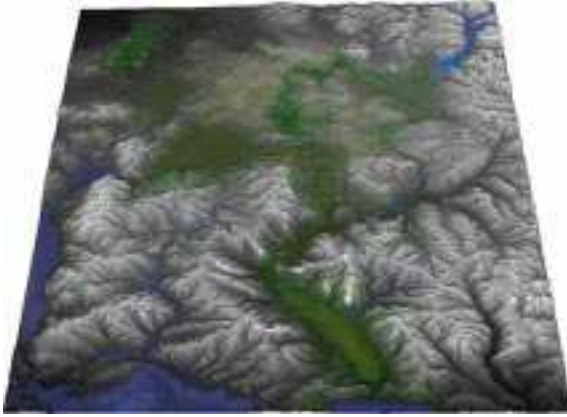
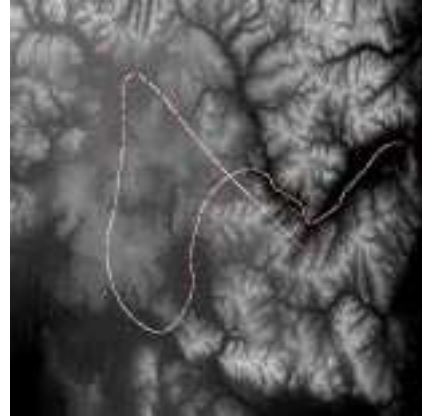


Fig. 3.8, page 146 : illumination probabiliste. Fig. 3.8, page 146 : déformation d'un arbre multi-texture.

### Planches couleurs 4 : paysages



*Vue globale du terrain, un coté mesure 10 km.*



*Représentation des altitudes  
et de la trajectoire de la caméra*



*Vue du paysage intégrant nuages, reliefs, arbres et ombres  
(15 minutes de calcul).*



*Autre point de vue pour la même description.  
On remarque l'effet du clipping par  
le module d'interprétation.*



*Autre vue du paysage intégrant tous les types d'éléments  
(environ 30 minutes de calcul).*

Neuvième partie

**Bibliographie**



# Bibliographie

- [AK84] M. Aono and T. L. Kunii. Botanical tree image generation. *IEEE Computer Graphics and Applications*, 4(5):10–34, May 1984.
- [Ake93] Kurt Akeley. RealityEngine graphics. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 109–116, August 1993.
- [AM90] Anthony A. Apodaca and M. W. Mantle. Renderman: Pursuing the future of graphics. *IEEE Computer Graphics and Applications*, 10(4):44–49, July 1990.
- [App68] Arthur Appel. Some techniques for shading machine renderings of solids. In *AFIPS 1968 Spring Joint Computer Conf.*, volume 32, pages 37–45, 1968.
- [Bli82] J. F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. In *Computer Graphics (SIGGRAPH '82 Proceedings)*, volume 16, pages 21–29, July 1982.
- [Blo85] Jules Bloomenthal. Modeling the mighty maple. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 305–311, July 1985.
- [Bou70] W. Jack Bouknight. A procedure for generation of three-dimensional half-toned computer graphics presentations. *Communications of the ACM*, September 1970.
- [Car84] Loren Carpenter. The A-buffer, an antialiased hidden surface method. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 103–108, July 1984.
- [Cat74] Edwin E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, Dept. of CS, U. of Utah, December 1974.
- [CCC87] Robert L. Cook, Loren Carpenter, and Edwin Catmull. The Reyes image rendering architecture. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, pages 95–102, July 1987.



- [Cha95] Christophe Chaudy. Systèmes de particules pour la modélisation et le rendu d'objets naturels. In *Actes des 3ème journées de l'AFIG*, pages 311–318, November 1995.
- [Cha96] Christophe Chaudy. Systèmes de particules pour la modélisation et le rendu d'objets naturels. *Revue internationale de CFAO et d'informatique graphique*, 11(4):455–466, 1996. ISSN 0298-0924.
- [CORLS95] D. Cohen-Or, E. Rich, U. Lerner, and V. Shenkar. Real-time photo-realistic visual flythrough. TR TR 301/95, School of Mathematical Sciences, Tel-Aviv university, July 1995.
- [CS93] Daniel Cohen and Amit Shaked. Photo-realistic imaging of digital terrains. In R. J. Hubbard and R. Juan, editors, *Eurographics '93*, pages 363–373, Oxford, UK, 1993. Eurographics, Blackwell Publishers.
- [CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, January 1982.
- [dBD95] Mark de Berg and Katrin T. G. Dobrindt. On levels of detail in terrains. TR UU-CS-1995-12, Department of Computer Science, Utrecht University, April 1995.
- [DG95] Mathieu Desbrun and Marie-Paule Gascuel. Animating soft substances with implicit surfaces. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 287–290. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.
- [DNKY95] Yoshinori Dobashi, Tomoyuki Nishita, Kazufumi Kaneda, and Hideo Yamashita. Fast display method of sky color using basis functions. In *Pacific Graphics '95*, August 1995.
- [dREF<sup>+</sup>88] Phillippe de Reffye, Claude Edelin, Jean Francon, Marc Jaeger, and Claude Puech. Plant models faithful to botanical structure and development. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 151–158, August 1988.
- [eNJ92] D. Arquès et N. Janey. Modélisation de cartes planaires pour la synthèse d'image de reliefs montagneux. In *11èmes Conférence internationale sur la CFAO, l'infographie et les technologies assistées par ordinateur, MICAD'92*, February 1992.
- [FFC82] A. Fournier, D. Fussell, and L. Carpenter. Computer rendering of stochastic models. *Communications of the ACM*, 25(6):371–384, June 1982.

- [FKN80] H. Fuchs, Z. M. Kedem, and B. F. Naylor. On visible surface generation by a priori tree structures. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 14, pages 124–133, July 1980.
- [Fou89] Alain Fournier. The modelling of natural phenomena. In *Proceedings of Graphics Interface '89*, pages 191–202, June 1989.
- [Gar84] Geoffrey Y. Gardner. Simulation of natural scenes using textured quadric surfaces. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 11–20, July 1984.
- [Gar85] Geoffrey Y. Gardner. Visual simulation of clouds. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 297–303, July 1985.
- [GH95] Michael Garland and Paul Heckbert. Fast polygonal approximation of terrains and height fields. TR CMU-CS-95-181, Computer Science Department, Carnegie Mellon University, September 1995.
- [GK93] Ned Greene and M. Kass. Hierarchical Z-buffer visibility. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 231–240, 1993.
- [Gla84] Andrew S. Glassner. Space subdivision for fast ray tracing. *IEEE Computer Graphics and Applications*, 4(10):15–22, October 1984.
- [Gre86a] Mark Green. A survey of three dialogue models. *ACM Transactions on Graphics*, 5(3):244–275, 1986.
- [Gre86b] Ned Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11), November 1986.
- [Gre89] Ned Greene. Voxel space automata: Modeling with stochastic growth processes in voxel space. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 175–184, July 1989.
- [GTGB84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaille. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 212–22, July 1984.
- [HA90] Paul E. Haeberli and Kurt Akeley. The accumulation buffer: Hardware support for high-quality rendering. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 309–318, August 1990.
- [HG94] Paul Heckbert and Michael Garland. Multiresolution modeling for fast rendering. In *Proceedings of Graphics Interface '94*, pages 43–50, Banff, Alberta, Canada, May 1994. Canadian Information Processing Society.

- [HL90] Pat Hanrahan and Jim Lawson. A language for shading and lighting calculations. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 289–298, August 1990.
- [Hon71] H. Honda. Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology*, 31:331–338, 1971.
- [HTSG91] Xiao D. He, Kenneth E. Torrance, Francois X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 175–186, July 1991.
- [Kaj85] James T. Kajiya. Anisotropic reflection models. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 15–21, July 1985.
- [Kaw82] K. Kawaguchi. A morphological study of the form of nature. In *Computer Graphics (SIGGRAPH '82 Proceedings)*, volume 16, pages 223–232, July 1982.
- [KV84] James T. Kajiya and Brian P. Von Herzen. Ray tracing volume densities. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 165–174, July 1984.
- [LD96] Bernd Lintermann and Olivier Deussen. Interactive modelling and animation of branching botanical structures. In *Eurographics '96*, pages 139–251. Eurographics, 1996.
- [LJF<sup>+</sup>91] A. Luciani, S. Jimenez, J. L. Florens, C. Cadoz, and O. Raoult. Computational physics: A modeler-simulator for animated physical objects. In Werner Purgathofer, editor, *Eurographics '91*, pages 425–436. North-Holland, September 1991.
- [LKR<sup>+</sup>96] Peter Lindstrom, David Koller, William Ribarsky, Larry F. Hodges, and Nick Faust. Real-time, continuous level of detail rendering of height fields. In *Eurographics '96*, volume 30, August 1996.
- [LP95] Jean-Christophe Lombardo and Claude Puech. Oriented particles: A tool for shape memory objects modelling. In *Proceedings of Graphics Interface '95*, Quebec city, Canada, May 1995. Canadian Information Processing Society.
- [Man83] B. B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman and Co., New York, rev 1983.

- [MC96] Alexandre Masselot and Bastien Chopard. Lattice gas modeling of snow transport by wind. Article disponible sur <http://cuiwww.unige.ch/masselot/>, May 1996.
- [McM95] Leonard McMillan. A list-priority rendering algorithm for redisplaying projected surfaces. TR TR95-005, University of North Carolina, 1995.
- [MGP<sup>+</sup>96] D. M. McKeown, S. J. Gifford, M. F. Polis, J. McMahon, , and C. D. Hoffman. Progress in automated virtual worlds construction. TR CMU-CS-96-102, Computer Science Department, Carnegie Mellon University, January 1996.
- [Mil86] Gavin S. P. Miller. The definition and rendering of terrain maps. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 39–48, August 1986.
- [MKM89] F. Kenton Musgrave, Craig E. Kolb, and Robert S. Mace. The synthesis and rendering of eroded fractal terrains. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 41–50, July 1989.
- [MM89] F. Kenton Musgrave and Benoit B. Mandelbrot. Natura ex machina. *IEEE Computer Graphics and Applications*, 9(1):4–7, January 1989.
- [Mus90] F. Kenton Musgrave. Natura ex machina II. *IEEE Computer Graphics and Applications*, 10(6):5–7, November 1990.
- [NDKY96] T. Nishita, Yoshinori Dobashi, Kazufumi Kaneda, and Hideo Yamashita. Display method of the sky color taking into account multiple scattering. In *Pacific Graphics '96*, pages 117–132, August 1996.
- [Ney96] Fabrice Neyret. *Textures Volumiques pour la Synthèse d'Images*. PhD thesis, Université de Paris XI Orsay, June 1996.
- [NMN87] Tomoyuki Nishita, Yasuhiro Miyawaki, and Eihachiro Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 303–310, July 1987.
- [NNS72] Martin E. Newell, R. G. Newell, and T. L. Sancha. A solution to the hidden surface problem. In *Proc. ACM Nat. Mtg.* 1972.
- [NO95] Max Nelson and Keiichi Ohsaki. Rendering trees from precomputed z-buffer views. In *Proceedings of the 6th Eurographics Workshop on Rendering*, June 1995.
- [NSTN93] Tomoyuki Nishita, Takao Sirai, Katsumi Tadamura, and Eihachiro Nakamae. Display of the earth taking into account atmospheric scattering. In James T.

- Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 175–182, August 1993.
- [NYE96] T. Nishita, Y. Dobashi, and E. Nakamae. Display of clouds taking into account multiple anisotropic scattering and sky light. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 379–386, August 1996.
- [Opp86] Peter E. Oppenheimer. Real time design and animation of fractal plants and trees. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 55–64, August 1986.
- [PHM93] Przemyslaw Prusinkiewicz, Mark S. Hammel, and Eric Mjolsness. Animation of plant development. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 351–360, August 1993.
- [Pho75] Bui-T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.
- [PJM94] Przemyslaw Prusinkiewicz, Mark James, and Radomiř Měch. Synthetic topiary. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 351–358. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [PL90] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990.
- [PLH88] Przemyslaw Prusinkiewicz, Aristid Lindenmayer, and James Hanan. Developmental models of herbaceous plants for computer imagery purposes. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 141–150, August 1988.
- [Pru93] Przemyslaw Prusinkiewicz. Modelling and visualization of biological structures. In *Proceedings of Graphics Interface '93*, pages 128–137, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.
- [RB85] William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 313–322, July 1985.
- [Ree83] W. T. Reeves. Particle systems – a technique for modeling a class of fuzzy objects. *ACM Trans. Graphics*, 2:91–108, April 1983.

- [Ric95] Jonathan Rice. Spatial bounding of iterated function system attractor sets. TR RT 132, Laboratoire ARTEMIS-IMAG, Université J. Fourier, Grenoble, March 1995.
- [Rob63] Lawrence G. Roberts. Machine perception of three-dimensional solids. TR 315, Lincoln Lab, MIT, Lexington, MA, May 1963.
- [RPP93] P. Roudier, B. Peroche, and M. Perrin. Landscapes synthesis achieved through erosion and deposition process simulation. In R. J. Hubbard and R. Juan, editors, *Eurographics '93*, pages 375–383, Oxford, UK, 1993. Eurographics, Blackwell Publishers.
- [RSC87] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering anti-aliased shadows with depth maps. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 283–291, July 1987.
- [RW94] Todd Reed and Brian Wyvill. Visual simulation of lightning. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 359–364. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [Sim90] Karl Sims. Particle animation and rendering using data parallel computation. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 405–413, August 1990.
- [SKv<sup>+</sup>92] Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, and Paul E. Haberli. Fast shadows and lighting effects using texture mapping. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 249–252, July 1992.
- [Smi84] Alvy Ray Smith. Plants, fractals and formal languages. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 1–10, July 1984.
- [SPS94] P. Slusallek, T. Pflaum, and H-P. Seidel. Implementing RenderMan - practice, problems and enhancements. In *Computer Graphics Forum*, volume 13, pages 443–454. Eurographics, Basil Blackwell Ltd, 1994. Eurographics '94 Conference issue.
- [TG96] J.M Trivi and Ph. Genoud. Spécification interactive de scénarios d'applications par atn. intégration à un environnement de programmation visuelle. In *5èmes journées internationales de Montpellier: Interface des Mondes Reels et Virtuels*, May 1996.

- [Ups89] Steve Upstill. *The RenderMan Companion: A Programmer's Guide to Realistic Computer Graphics*. Addison-Wesley. ISBN 0-201-50868-0, 1989.
- [VEJA89] Xavier Gérard Viennot, Georges Eyrolles, Nicolas Janey, and Didier Arqués. Combinatorial analysis of ramified patterns and computer imagery of trees. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 31–40, July 1989.
- [Vos85] R. P. Voss. Fractal forgeries. In R. A. Earnshaw, editor, *Fundamental Algorithms for Computer Graphics*. Springer-Verlag, 1985.
- [Woo70] W. A. Woods. Transition network grammars for natural language analysis. *Communication of the ACM*, 13(10):591–606, 1970.
- [WP95] Jason Weber and Joseph Penn. Creation and rendering of realistic trees. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 119–128. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.
- [WW92] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques: Theory and Practice*. Addison-Wesley Publishing Company, 1992.





MODÉLISATION ET RENDU D'IMAGES RÉALISTES  
DE PAYSAGES NATURELS

RÉSUMÉ

La synthèse d'images de paysages naturels reste encore un problème peu abordé en infographie. Pourtant, des modèles convaincants en particulier pour la modélisation de végétaux existent, mais n'ont pu être exploités dans des scènes comportant un nombre très important d'éléments. Je présente dans ce travail, une approche pour la modélisation et le rendu d'objets naturels qui doit permettre de contourner l'obstacle de la complexité géométrique et offrir un cadre simple pour la création de structures naturelles. Ainsi un modèle basé sur les systèmes de particules est proposé pour la création de structures végétales. Ce modèle fait appel à la définition d'un automate pour le contrôle des trajectoires, de la création et de la destruction des particules. Il peut être également utilisé pour modéliser et rendre un certain nombre d'autres phénomènes naturels (nuages, eau...). Par ailleurs, une scène de paysage naturel se compose de très nombreux éléments ayant des échelles très différentes sur l'image finale. Aussi nous proposons des techniques rapides et précises, basées sur les multi-textures, qui permettent un rendu multi-échelles efficace des éléments de végétation. De part leurs caractéristiques, ces méthodes peuvent être intégrées dans une architecture spécialisée pour la synthèse d'images de paysages naturels que nous présentons.

SUMMARY

Today, realistic image synthesis of natural landscapes still remains a challenge in the field of computer graphics. Although convincing visual models of plants exist, they cannot be integrated in very large scenes with numerous natural elements. We present a model to create and render several types of fuzzy objects such as trees, grass and clouds. This model bypasses the geometric complexity problem and offers facilities to define natural structures. Particle systems are used to build the object geometry and a finite state automata to control the particle behaviour. So, the branch geometry, the leaf position and other geometric parameters are defined by the particle trajectory in the 3D space.

Furthermore, a natural landscape scene may be composed of many elements, we propose fast and precise methods, based on multi-texture representation, to efficiently render multi-resolution vegetation elements.

All these models are simple and fast and thus, very suitable to be integrated in a natural landscape rendering system. We present such a specialized architecture dedicated to the rendering of realistic images over an heterogeneous computer network.