

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mohamed Khider Biskra  
Faculté des Sciences Exactes, des sciences de la Nature et de la Vie  
Département d'Informatique

N° d'ordre :.....  
Série :.....



## Mémoire

Présenté en vue de l'obtention du diplôme de Magister en Informatique

Option: Synthèse d'images et vie artificielle

Titre :

# REPRESENTATION A BASE D'IMPOSTEURS POUR LE RENDU TEMPS REEL DES FORETS

Par :

**ABBAS Fayçal**

Soutenu le : 20-09-2011

### Devant le jury :

<b>Pr. N. DJEDI</b>	<b>PR</b>	<b>Université de Biskra</b>	<b>Président</b>
<b>Dr. M.C.BABAHENINI</b>	<b>MCA</b>	<b>Université de Biskra</b>	<b>Rapporteur</b>
<b>Dr. F. CHERIF</b>	<b>MCA</b>	<b>Université de Biskra</b>	<b>Examineur</b>
<b>Dr. A. ZIDANI</b>	<b>MCA</b>	<b>Université de Batna</b>	<b>Examineur</b>

---

# Remerciements

---

Je tiens à remercier sincèrement Monsieur Dr. **Med Chaouki BABAHENINI**, d'avoir accepté de m'encadrer, J'ai beaucoup bénéficié de ses précieux conseils, de ses suggestions pertinentes pour la réalisation de ce travail, tout en me laissant une liberté que j'ai apprécié.

Je tiens également à remercier les membres du jury qui m'ont fait l'honneur de bien vouloir évoluer mon travail, et plus précisément :

Pr. **DJEDI NourEddine** pour l'honneur qu'il me fait, en acceptant la présidence de ce jury.  
Dr. Monsieur **CHERIF Foudil**, Dr. **Zidani Abdelmadjid** auxquels je suis très reconnaissant d'avoir accepté d'être examinateurs de ce travail.

Je remercie en particulier : Tous les membres de ma famille.

Et finalement, je n'oublis pas à remercier tous ceux qui ont participé directement ou indirectement à l'aboutissement de ce travail.

---

# Table de matières

---

<b>1</b>	<b>INTRODUCTION .....</b>	<b>5</b>
<b>2</b>	<b>COMPLEXITE EN SYNTHÈSE D'IMAGE : .....</b>	<b>8</b>
<b>3</b>	<b>RENDU A BASE D'IMAGE : .....</b>	<b>9</b>
3.1	REPRESENTATION DES SCENES STATIQUES .....	9
3.1.1	<i>Les représentations sans géométrie .....</i>	<i>10</i>
3.1.2	<i>Utilisation de la géométrie de manière implicite .....</i>	<i>16</i>
3.1.3	<i>Utilisation de la géométrie de manière Explicite .....</i>	<i>18</i>
3.2	REPRESENTATION DES SCENES DYNAMIQUES .....	32
3.3	TECHNIQUES HYBRIDES .....	35
3.3.1	<i>Le placage de texture .....</i>	<i>35</i>
3.3.2	<i>Placage de texture en relief .....</i>	<i>36</i>
<b>4</b>	<b>CONCLUSION .....</b>	<b>38</b>
	<b>REPRESENTATION DES FORETS EN SYNTHÈSE D'IMAGE .....</b>	<b>40</b>
<b>1</b>	<b>INTRODUCTION .....</b>	<b>40</b>
<b>2</b>	<b>REPRESENTATION D'ARBRE EN SYNTHÈSE D'IMAGE .....</b>	<b>41</b>
2.1	LES MODELES STRUCTURELES .....	41
2.1.1	<i>Les premiers modèles à caractère géométrique .....</i>	<i>41</i>
2.1.2	<i>Les modèles à base de paramètres botaniques .....</i>	<i>43</i>
2.1.3	<i>Les modèles à base de grammaires .....</i>	<i>44</i>
2.1.4	<i>Les modèles combinatoires .....</i>	<i>46</i>
2.2	LES MODELES IMPRESSIONNISTES .....	47
2.2.1	<i>Les modèles à base de textures .....</i>	<i>47</i>
2.2.2	<i>Les modèles dans un espace de voxels .....</i>	<i>48</i>
2.2.3	<i>Les modèles à base de système de particules .....</i>	<i>49</i>
<b>3</b>	<b>BILAN ET CONCLUSION .....</b>	<b>50</b>

<b>REPRESENTATION A BASE DE BILLBOARDS DES FORETS EN TEMPS REEL .....</b>	<b>53</b>
<b>1 INTRODUCTION .....</b>	<b>53</b>
<b>2 LES TECHNIQUES D'IMPOSTEUR TEMPS REEL POUR LES FORETS .....</b>	<b>53</b>
2.1 LES APPROCHES TEXTURES .....	53
2.2 MODELE HIERARCHIQUE POUR LE RENDU INTERACTIF DE FORETS .....	54
2.3 LES NUAGES DE PANNEAUX D'AFFICHAGE (BILLBORAD CLOUDS).....	56
2.4 BILLBOARDS VOLUMIQUES.....	57
2.5 RENDU REALISTE DE SCENES NATURELLES .....	59
<b>3 BILAN.....</b>	<b>62</b>
<b>LE MODELE PROPOSE : GENERATION DES FORETS EN TEMPS REEL .....</b>	<b>65</b>
<b>1 INTRODUCTION ET MOTIVATIONS.....</b>	<b>65</b>
<b>2 DESCRIPTION DU MODELE PROPOSE .....</b>	<b>66</b>
2.1 GENERATION DES ARBRES AVEC LA TECHNIQUE DE BILLBOARD .....	66
2.1.1 <i>Placage de texture et transparence.....</i>	<i>68</i>
2.1.2 <i>Calcul de la transformation .....</i>	<i>70</i>
2.2 GENERATION DES ARBRES AVEC LES L-SYSTEME.....	73
2.2.1 <i>Représentation d'arbre avec un L-système.....</i>	<i>73</i>
2.2.2 <i>Génération de la structure de l'arbre.....</i>	<i>74</i>
2.2.3 <i>Interprétation géométrique .....</i>	<i>76</i>
2.2.4 <i>Génération des feuilles de l'arbre .....</i>	<i>77</i>
<b>3 CONCLUSION .....</b>	<b>80</b>
<b>MISE EN ŒUVRE, RESULTATS ET BILAN .....</b>	<b>82</b>
<b>1 INTRODUCTION .....</b>	<b>82</b>
<b>2 LES ALGORITHMES UTILISES .....</b>	<b>82</b>
<b>3 RESULTATS OBTENUS .....</b>	<b>86</b>
<b>4 DISCUSSION.....</b>	<b>94</b>
<b>5 LIMITATION DU MODELE .....</b>	<b>94</b>
<b>6 CONCLUSION .....</b>	<b>95</b>
<b>CONCLUSION ET PERSPECTIVES .....</b>	<b>96</b>
<b>BIBLIOGRAPHIE.....</b>	<b>98</b>

---

# Liste de figure

---

<b>Figure 1.</b> <i>fonction Plénoptique.</i> .....	<b>11</b>
<b>Figure 2.</b> <i>Camera plénoptique.</i> .....	<b>12</b>
<b>Figure 3.</b> <i>vue panoramiques.</i> .....	<b>12</b>
<b>Figure 4.</b> <i>Light Field à 360 degrés.</i> .....	<b>13</b>
<b>Figure 5.</b> <i>Fonctionnement du Light field : (a) principe de l'acquisition de chaque rayon des images dans le plan (u, v) avec un exemple d'imagette obtenue, (b) principe du rendu d'une nouvelle vue pour un seul élément discret du plan (r, s) avec un exemple d'échantillon obtenu.</i> .....	<b>14</b>
<b>Figure 6.</b> <i>Images panoramiques.</i> .....	<b>15</b>
<b>Figure 7.</b> <i>Prise de vues de la mosaïque concentrique.</i> .....	<b>15</b>
<b>Figure 8.</b> <i>Nouvelle scène obtenu par Mosaïques concentriques</i> .....	<b>16</b>
<b>Figure 9.</b> <i>View morphing.</i> .....	<b>17</b>
<b>Figure 10.</b> <i>Un morphe déformant une forme.</i> .....	<b>18</b>
<b>Figure 11.</b> <i>Exemple de sprite.</i> .....	<b>18</b>
<b>Figure 12.</b> <i>(a) : billboard face à la caméra, (b) billboard vue de coté, (c) billboard vue dessus.</i> .....	<b>20</b>
<b>Figure 13.</b> <i>(a) billboards face au plan image de la caméra, (b) billboards orientés vers le centre de la caméra.</i> .....	<b>20</b>
<b>Figure 14.</b> <i>Billboard en croix.</i> .....	<b>21</b>
<b>Figure 15.</b> <i>Etapas de "tour into the picture".</i> .....	<b>22</b>
<b>Figure 16.</b> <i>Exemple billboard cloud.</i> .....	<b>23</b>
<b>Figure 17.</b> <i>Microfacet billboarding.</i> .....	<b>24</b>

<b>Figure 18.</b> <i>Un objet (a) et son imposteur, (b) L'imposteur est vue d'un point de vue différent que la géométrie originale.</i> .....	25
<b>Figure 19.</b> <i>Imposteurs en couches.</i> .....	26
<b>Figure 20.</b> <i>Imposteurs maillés de Sillion.</i> .....	28
<b>Figure 21.</b> <i>3D image warping.</i> .....	28
<b>Figure 22.</b> <i>Réalité virtualisée.</i> .....	33
<b>Figure 23.</b> <i>Mur de caméras</i> .....	34
<b>Figure 24.</b> <i>Placage de texture.</i> .....	35
<b>Figure 25.</b> <i>Exemples de reliefs produits par Oliveira [OBM00].</i> .....	37
<b>Figure 26.</b> <i>(a) Objet représenté par six textures en relief, (b) vue de l'objet reconstruit par placage de texture en relief.</i> .....	37
<b>Figure 27.</b> <i>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].</i> .....	42
<b>Figure 28.</b> <i>Exemples de plantes générées par un modèle à base de paramètres botaniques.</i> .....	43
<b>Figure 29.</b> <i>Construction du flocon de neige. A chaque itération, les segments sont remplacés par le générateur.</i> .....	44
<b>Figure 30.</b> <i>Exemple simple de L-système dont l'axiome est A. On peut assimiler A à un bourgeon et S à un segment d'entre-noeud.</i> .....	45
<b>Figure 31.</b> <i>Structures stochastiques avec branchements.</i> .....	45
<b>Figure 32.</b> <i>Deux L-systèmes construits par une application récursive de la même règle.</i> .....	46
<b>Figure 33.</b> <i>Exemple de représentation d'arbre avec une matrice de ramification.</i> .....	46
<b>Figure 34.</b> <i>Deux arbres générés par la méthode des matrices de ramification [VEJA89].</i> .....	47
<b>Figure 35.</b> <i>Billboard en croix pour donner une impression de volume.</i> .....	48
<b>Figure 36.</b> <i>(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].</i> .....	49
<b>Figure 37.</b> <i>Un exemple de paysage généré par des systèmes de particules.</i> .....	50
<b>Figure 38.</b> <i>(a) billboard classique, (b) 2 billboard en croix, (c) 3 billboard croix.</i> .....	54

<b>Figure 39.</b> <i>Construction d'une BTF.</i> .....	<b>55</b>
<b>Figure 40.</b> <i>Interpoler de manière pondérée des images afin d'obtenir une représentation proche du billboard.</i> .....	<b>56</b>
<b>Figure 41.</b> <i>Exemple d'un nuage de panneaux : (a) modèle original (1,960 polygones) (b) rendu en utilisant une couleur par panneau (c) Vue de 52 panneaux texturés (d) nuage de panneaux rendu.</i> .....	<b>56</b>
<b>Figure 42.</b> <i>Un volume mip-map construit à partir de 6 directions de vue.</i> .....	<b>58</b>
<b>Figure 43.</b> <i>Arbres présentés par la méthode : Billboard volumique [DF09].</i> .....	<b>59</b>
<b>Figure 44.</b> <i>Les trois niveaux de détail, choisis en fonction de la distance de l'observateur.</i> .....	<b>60</b>
<b>Figure 45.</b> <i>(a) Brins d'herbe avec chaînes de quadrilatères texturés. (b) Scan d'un brin</i> .....	<b>60</b>
<b>Figure 46.</b> <i>Densité variable de brins définie par une carte de densité. L'image de gauche est une carte qui permet d'obtenir le résultat de l'image de droite.</i> .....	<b>61</b>
<b>Figure 47.</b> <i>(a) Paramétrisation de l'enveloppe d'un arbre depuis la feuille centrée en P.</i> .....	<b>62</b>
<b>Figure 48.</b> <i>Forêt présentée par la méthode de KEVIN [Kev08].</i> .....	<b>62</b>
<b>Figure 49.</b> <i>La structure de billboard utiliser.</i> .....	<b>67</b>
<b>Figure 50.</b> <i>Plusieurs billboard croisés.</i> .....	<b>67</b>
<b>Figure 51.</b> <i>Le placage de texture.</i> .....	<b>68</b>
<b>Figure 52.</b> <i>Le placage de texture d'arbre sur le billboard.</i> .....	<b>69</b>
<b>Figure 53.</b> <i>Billboards avec transparence.</i> .....	<b>70</b>
<b>Figure 54.</b> <i>Extraction d'information.</i> .....	<b>71</b>
<b>Figure 55.</b> <i>Remplacement de la sous matrice par l'identité.</i> .....	<b>71</b>
<b>Figure 56.</b> <i>Représentation de billboard par rapport au caméra.</i> .....	<b>72</b>
<b>Figure 57.</b> <i>Rendu une forêt de 1140 arbres, chaque arbre représentée par un billboard plat.</i> .....	<b>72</b>
<b>Figure 58.</b> <i>Une scène de 1600 arbres, chaque arbre représentée par 8 billboards en croix.</i> .....	<b>73</b>
<b>Figure 59.</b> <i>Processus de construction de squelette à partir d'un L-système.</i> .....	<b>75</b>
<b>Figure 60.</b> <i>Interprétation géométrique d'une squelette.</i> .....	<b>76</b>
<b>Figure 61.</b> <i>Interprétation géométrique d'un squelette avec la primitive cylindre.</i> .....	<b>76</b>
<b>Figure 62.</b> <i>Ecorce et branches d'un arbre texturé avec un placage de textures 2D.</i> .....	<b>77</b>
<b>Figure 63.</b> <i>Quelques exemples de texture 2D, illustrant la diversité des feuilles.</i> .....	<b>78</b>

<b>Figure 64.</b> <i>Deux exemples d'arbres (a ) et (b) générées par notre système. ....</i>	<b>79</b>
<b>Figure 65.</b> <i>Deux arbres avec des textures d'écorce et feuilles différent. ....</i>	<b>80</b>
<b>Figure 66.</b> <i>Remplacement de la sous matrice par l'identité. ....</i>	<b>83</b>
<b>Figure 67.</b> <i>Une scène de 1600 arbres, chaque arbre représentée par 8 billboards en croix.....</i>	<b>86</b>
<b>Figure 68.</b> <i>Une scène de 1600 arbres, chaque arbre représentée par 8 billboards en croix.....</i>	<b>87</b>
<b>Figure 69.</b> <i>Une scène de 3600 arbres chaque arbre représentée par 1 billboard plat. ....</i>	<b>87</b>
<b>Figure 70.</b> <i>Comparaison entre un modèle Billboard plat et Billboard en croix.....</i>	<b>88</b>
<b>Figure 71.</b> <i>Arbre rendu avec un nombre différents de billboards. ....</i>	<b>89</b>
<b>Figure 72.</b> <i>Comparaison entre un modèle géométrique pur et un modèle hybride .....</i>	<b>90</b>
<b>Figure 73.</b> <i>Comparaison entre un modèle géométrique pur et le modèle hybride .....</i>	<b>91</b>
<b>Figure 74.</b> <i>Une forêt de 400 arbres.....</i>	<b>92</b>
<b>Figure 75.</b> <i>Une forêt de 100 arbres.....</i>	<b>92</b>
<b>Figure 76.</b> <i>Une forêt de 100 arbres avec éclairage et ombrage. ....</i>	<b>93</b>
<b>Figure 77.</b> <i>Une forêt de 520 : 120 arbres billboard-Lsystème, 120 arbres billboard croisé, 280 billboard plats. ....</i>	<b>93</b>

---

# Liste de Tables

---

**Tableau 1.** Matrice géométrique - rendu [SRC96].

**Tableau 2.** Caractéristique des techniques de champ de lumière.

**Tableau 3.** Comparaison des techniques d'imposteurs.

---

# Résumé

---

La représentation et la visualisation d'une forêt en temps réel est une tâche difficile, la principale difficulté réside dans la complexité des scènes générées, qui contiennent une quantité très importante de petits détails difficiles à modéliser, coûteux à rendre et qui nécessitent une quantité très importante de mémoire.

L'objectif de notre travail de recherche consiste à présenter 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 modèle simple pour la création de structures naturelles.

Notre travail se focalise sur les arbres qui représentent la forêt. Afin de simplifier la complexité de la scène, les arbres vue de loin sont représentées par des billboards (simple, croisés,...), pour les arbres vue de près nous proposons un modèle hybride entre billboards et L-systèmes, où nous séparons la représentation de l'arbre en deux entités distinctes :

le tronc et les branches sont représentées par une grammaire L-système et les feuilles sont représentées par des billboards, ceci augmente le réalisme des arbres proches par rapport au point de vue.

---

# Abstract

---

Representation and visualization of a forest in real time is a difficult task, the main difficulty of this problem is the complexity of forest scenes which contain a very large quantity of small details difficult to model, expensive to make and require very large amount of memory.

The aim of our research is to present an approach for modeling and rendering of natural objects that must be used to circumvent the obstacle of the geometric complexity and provide simple model for the creation of natural structures.

Our work focuses on trees that represent the forest. To simplify the complexity of the scene, the trees from a distance are represented by billboards (single, cross, ...) for lending tree view we propose an hybrid model between billboards and L-systems, where we separate the tree representation into two separate entities trunks and branches are represented by an L-system grammar and leaves are represented by billboards, this increases the realism of nearby trees from the point of view.

---

# ملخص

---

إن تمثيل وتصوير غابة في الوقت الحقيقي هي مهمة صعبة، والصعوبة الرئيسية في هذه المشكلة هي تعقيد مشاهد الغابات التي تحتوي على كمية كبيرة جداً من التفاصيل الصغيرة، هذه التفاصيل مكلفة في الوقت وتتطلب كمية كبيرة جداً من الذاكرة.

يهدف بحثنا إلى تقديم تقنية لوضع النماذج وإنشاء المشاهد الطبيعية التي تسمح بالتحايل على عقبة التعقيد الهندسي وتقديم نموذج بسيط لإنشاء المشاهد الطبيعية.

لتبسيط تعقيدات المشهد، نقوم بتمثيل الأشجار التي هي على مسافة بعيدة من نقطة النظر بلوحات (واحد، إثنان، ...) أما بالنسبة للأشجار القريبة فنقترح نموذج هجين بين اللوحات وأنظمة لام، أين نقوم بتقسيم تمثيل الشجرة إلى قسمين منفصلين الجذع نقوم بتمثيله بأنظمة لام أما الأوراق فنقوم بتمثيلها بتقنية اللوحات، وهذا يزيد من الواقعية بالنسبة للأشجار القريبة من نقطة النظر.

---

# Introduction générale

---

La modélisation et le rendu de scènes naturelles intéressent la communauté scientifique depuis plusieurs années en raison de ses nombreuses applications.

Les recherches sur le rendu ont permis le développement d'applications de visualisation interactives dans différents contextes : simulations d'écosystèmes, représentation 3D de données cartographiques, jeux vidéos, simulateurs de vol, etc.

De nos jours, les images générées par ordinateur deviennent de plus en plus réalistes, en effet, le but est de créer des images les plus proches possibles de ce qu'un observateur peut voir de ses yeux. La principale difficulté de ce problème est la complexité, les scènes naturelles contiennent une quantité très importante de petits détails difficiles à modéliser, coûteux à rendre et qui nécessitent une quantité très importante de mémoire.

Cette complexité provient surtout de la modélisation. La complexité géométrique provient du nombre très élevé de détail. Surmonter cette complexité a fait l'objet de nombreuses recherches. Nous avons étudié le problème en traitant le rendu d'arbres. Des algorithmes développés auparavant permettent déjà de rendre les plantes interactivement, cependant avec des approximations importantes. Il existe aussi des algorithmes beaucoup plus

réalistes mais qui nécessitent des temps de calcul très importants, en utilisant des méthodes de lancer de rayon par exemple. Même dans ce cas, des approximations doivent être employées pour que le rendu se fasse dans un temps raisonnable.

Le but de notre travail de recherche est le rendu des forêts en temps réel, notre travail se focalise sur les arbres qui représentent la forêt. Afin de simplifier la complexité de la scène, nous avons adopté une structure de multi représentation, ainsi les arbres vue de loin sont représentées par des billboards (simple, croisés,...), et pour les arbres vue de près nous proposons un modèle hybride entre billboards et L-systèmes.

Pour pouvoir rendre les forêts, ainsi modélisées en temps réel, il faut pouvoir utiliser au maximum la puissance de calcul des cartes graphiques actuelles (GPUs) et laisser le processeur central (CPU) libre pour effectuer d'autres tâches.

---

# Plan du mémoire

---

Tout d'abord nous commencerons par présenter un état de l'art des différentes techniques de rendu à base d'image. Dans ce chapitre nous essayons de définir très brièvement le principe de la synthèse d'images classique à base des modèles géométriques, comme nous citons quelques limitations et difficultés de ce type de synthèse. Puis nous présentons le principe général de la synthèse d'images à partir d'images qui est une alternative à l'approche classique.

Le chapitre 2 concerne la création des végétaux, nous verrons les méthodes employées pour générer les forêts et les techniques utilisées pour les rendre. Dans le chapitre suivant nous présenterons les techniques à base d'imposteurs pour le rendu de forêt en temps réel.

Nous aborderons ensuite la partie qui traite notre contribution où nous présenterons un modèle hybride entre les billboards et les L-systèmes pour la représentation et la visualisation d'une forêt en temps réel.

Enfin, le dernier chapitre est consacré à l'implémentation, l'évaluation et la présentation des résultats générés par l'application que nous avons développée, ce chapitre se termine par un ensemble de perspectives à atteindre dans des travaux futurs.

La conclusion générale récapitule cette représentation et met en évidence un ensemble de nouvelles voies à explorer ultérieurement pour enrichir ce travail.

**Chapitre 1 :**  
**RENDU A BASE D'IMAGE**

---

# RENDU A BASE D'IMAGE

---

## 1 Introduction

Les méthodes de rendu de synthèse d'images ne cessent de s'améliorer, tant en ce qui concerne les images photo-réalistes que les images générées en temps réel. On parvient ainsi à créer des films en images de synthèse dont le réalisme est parfois saisissant. Dans un même temps, les techniques de rendu d'images de synthèse en temps réel atteignent elles aussi un niveau de rendu impressionnant compte tenu des contraintes temps réel qui leur sont imposées. Ce constat est particulièrement frappant pour les jeux vidéo dont les créateurs ne cessent de chercher de nouvelles méthodes pour les rendre plus fluides et augmenter la sensation d'immersion.

Dans ce chapitre nous essayons de définir très brièvement la complexité de la synthèse d'images classique, comme nous citons quelques limites et difficultés de ce type de synthèse, puis nous présentons le principe général de la synthèse d'images à partir d'image qui est une alternative à l'approche classique. Nous avons classé les méthodes en fonction de leur indépendance croissante vis à vis de la géométrie, nous détaillerons particulièrement les imposteurs.

## 2 Complexité en synthèse d'image :

Rendre des scènes complexes en temps réel avec les méthodes de modélisation traditionnels pour l'industrie et en particulier pour les simulateurs ou les jeux vidéos est un problème immense du fait de la complexité et la quantité d'éléments qu'ils composent, comme par exemple une forêt ou une ville entière .....

Les approches classiques utilisent des modèles mathématiques pour représenter les objets composant la scène, cependant quelques objets ont des formes complexes comme par exemple arbres, chevelure, il est alors impossible de les afficher en temps réel à cause du :

1. Grand espace mémoire nécessaire.
2. Temps de calcul requis : Si la scène est trop complexe, calculer une seule image peut prendre plusieurs minutes.
3. Fait qu'une scène complexe demande une modélisation fastidieuse (un grand effort consacré pendant la phase de modélisation).

D'autres approches sont apparues pour résoudre ce problème : le rendu à base de point et le rendu à base d'image, d'où l'idée consistant à utiliser d'autres primitives pour augmenter la qualité de rendu sans augmenter la complexité géométrique de la scène.

Le rendu à base de points consiste à utiliser le point comme primitive de rendu et à modéliser un objet sous la forme d'un ensemble dense de points, chaque point doit être considéré indépendamment, et possède ces propres caractéristiques comme la couleur et la normale, les points peuvent être dessinés dans n'importe quel ordre, l'image se construisant au fur et à mesure qu'ils sont projetés, permis les avantages de cette technique est le fait qu'elle offre un rendu de haute qualité avec un temps de calcul très court, Le plus grand défaut est le coût de stockage extrêmement élevés. .

Le principe du rendu à base d'image consiste à utiliser l'image comme primitive de rendu pour réduire la complexité géométrique de la scène, il repose sur l'utilisation

uniquement des images est permet de générer de nouvelles images plus réalistes, avec un temps de rendu indépendant de la complexité géométrique de la scène, le problème de cette représentation est l'espace mémoire nécessaire et les difficultés pour varier l'éclairage de la scène.

Toutes ces limites motivent le rendu multi échelle qui consiste à considérer différentes représentations pour un même objet, et de choisir sa représentation au moment du rendu en fonction de sa distance à l'observateur. On affichera donc les versions les plus détaillées lorsque le modèle est proche de l'observateur et des versions plus en plus dégradées en fonction de l'éloignement de l'observateur.

### **3 Rendu à base d'image :**

En synthèse d'images, la dernière décennie a vu naître de nouveaux modèles pour lesquels des photographies d'objets réels ne sont plus seulement utilisées comme des textures indépendantes du point de vue. Elles servent de support pour des méthodes interactives de visualisation d'environnements ou d'objets réels complexes. Ces méthodes de visualisation, identifiées sous le nom de rendu basé-image permettent de réduire de manière drastique une description géométrique généralement très complexe tout en conservant un réalisme important.

#### **3.1 Représentation des scènes statiques**

Comme le montrent Shum [SRC96] dans un tour d'horizon des différentes techniques de rendu à base d'images. Il propose une classification des méthodes de rendu allant d'un côté du spectre les techniques de rendu sans description géométrique de la scène et de l'autre côté, les techniques où la géométrie de la scène est complètement définie. Entre ces deux extrêmes se trouvent les méthodes utilisant une description partielle, voire implicite de la géométrie de la scène, **Tableau 1** illustre quelques méthodes selon cette classification.

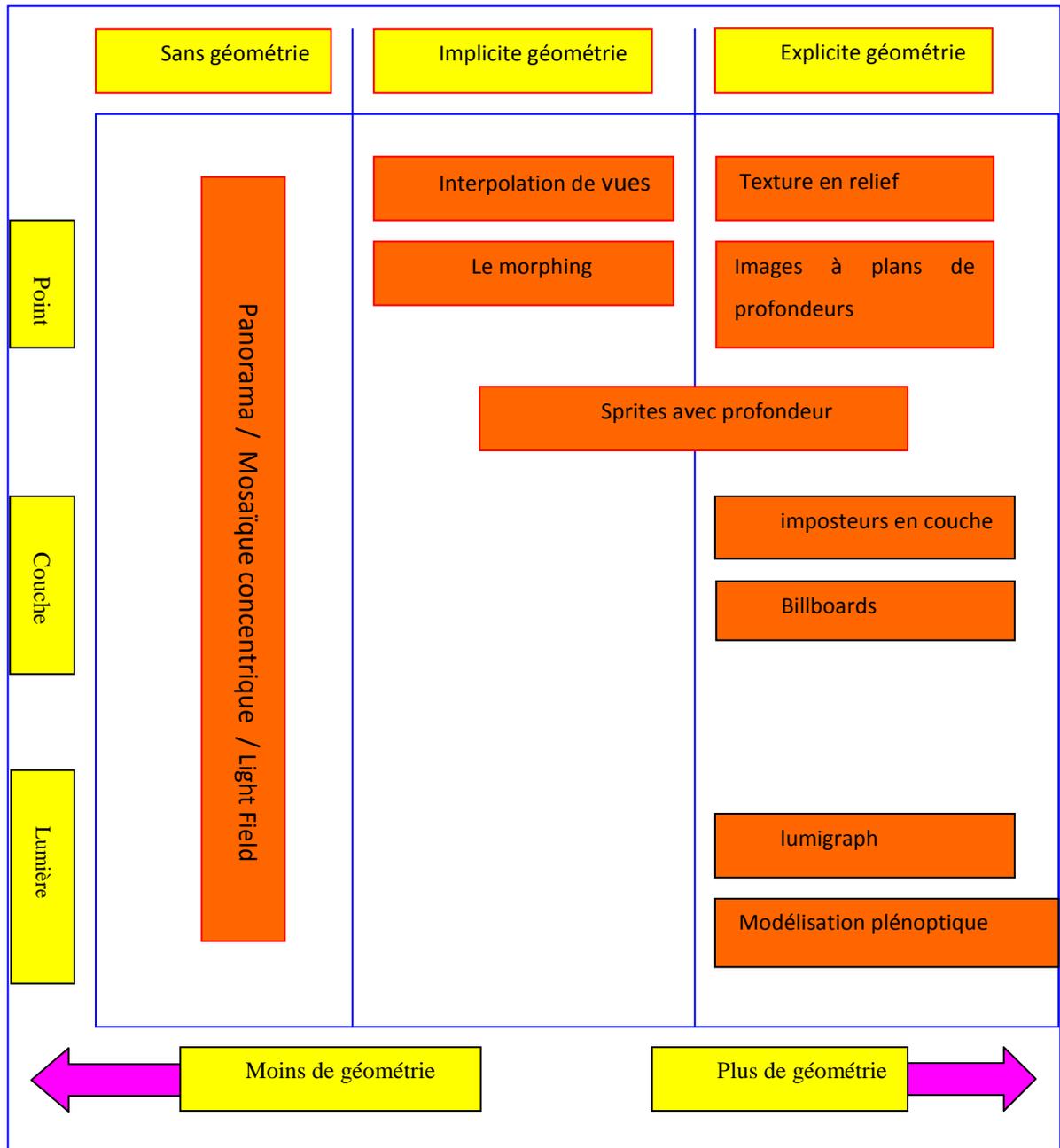


Tableau 3. Matrice géométrique - rendu [SRC96].

### 3.1.1 Les représentations sans géométrie

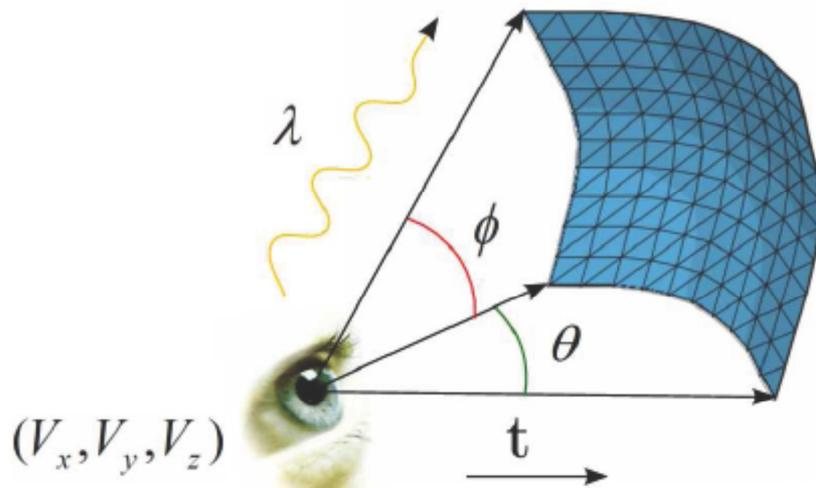
Ces techniques n'utilisent à aucun moment l'information géométrique présente dans la scène pour faire la reconstruction. Pour cela, ces techniques de reconstruction utilisent un autre moyen de représentation de l'environnement. Ce moyen est basé sur l'apparence et plus particulièrement sur l'intensité des rayons lumineux.

### 3.1.1.1 La fonction plénoptique

La description de l'environnement via des rayons lumineux est modélisée par une fonction appelée la fonction plénoptique. Cette fonction permet d'exprimer l'intensité d'un flux lumineux, qui passe au centre de la caméra, émis à partir de chaque point de l'espace et ce dans toutes les directions. La fonction plénoptique est proposée par E. Adelson et J. Bergen [AB91], elle est représentée à la figure 1.

La fonction plénoptique originale est définie selon 7 dimensions : pour tous les points de l'espace  $(V_x, V_y, V_z)$ , pour tous les angles possibles  $(\theta, \phi)$ , pour toutes les longueurs d'onde  $\lambda$ , et à tout instant  $t$ . La fonction plénoptique est de la forme

$$p = P(V_x, V_y, V_z, \theta, \phi, \lambda, t)$$



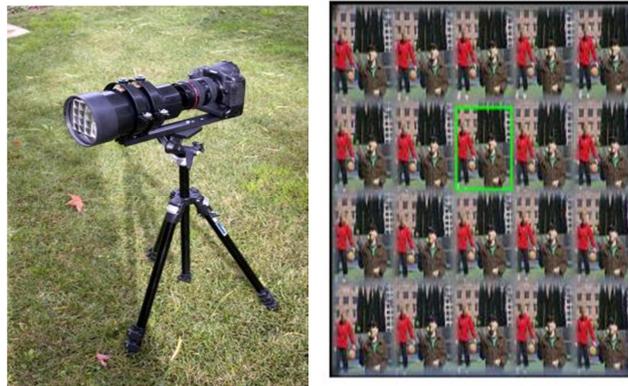
**Figure 1.** fonction Plénoptique.

Pour la plupart des travaux antérieurs en rendu par champ de lumière, on néglige les paramètres suivants :

- **Temps  $t$**  : on ne considère que des scènes statiques, et ou on suppose la propagation instantanée (vitesse infinie) de la lumière en régime permanent.
- **Longueur d'onde  $\lambda$**  : on traite séparément trois longueurs d'onde principales, correspondant aux trois longueurs d'onde associées aux phosphores d'un écran cathodique (c.a.d canaux R, G et B) ;
- **Phase, Cohérence** : ces paramètres ont en général peu d'effet sur l'image finale.

### 3.1.1.2 Modélisation plénoptique :

Dans [MB95], McMillan et Bishop choisissent d'ignorer le temps  $t$  et la longueur d'onde  $\lambda$  et d'introduire le plénoptique modeling qui est une fonction à 5 dimensions :



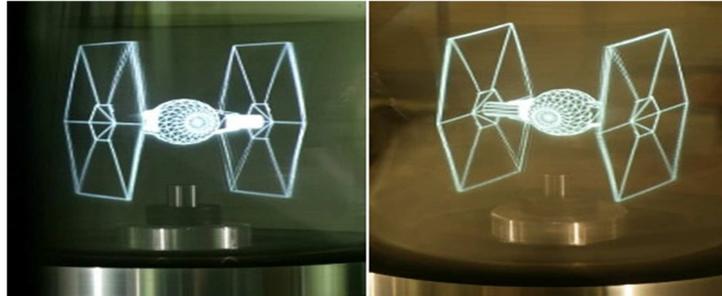
**Figure 2.** *Camera plénoptique.*

Ils enregistrent une scène statique en plaçant des caméras dans l'espace scène 3D, chacun sur un trépied permettant un mouvement vertical et de rotation. À chaque position, une image projetée cylindriquement est composée à partir des images capturées pendant une rotation panoramique. Cette projection cylindrique est obtenue par la reconstruction de la fonction plénoptique qui nécessite d'estimer le flot lumineux à chaque position d'une caméra. Pour cela, à partir de deux images (ou plus) de référence panoramiques projetées cylindriquement, le champ lumineux est calculé par disparité stéréoscopique. La projection cylindrique à un nouveau point de vue peut être obtenue par un placage de type cylindre cylindre, puis peut être reprojétée sur un plan permettant d'obtenir une image.



**Figure 3.** *vue panoramiques.*

### 3.1.1.3 Le Lightfield et le Lumigraph :

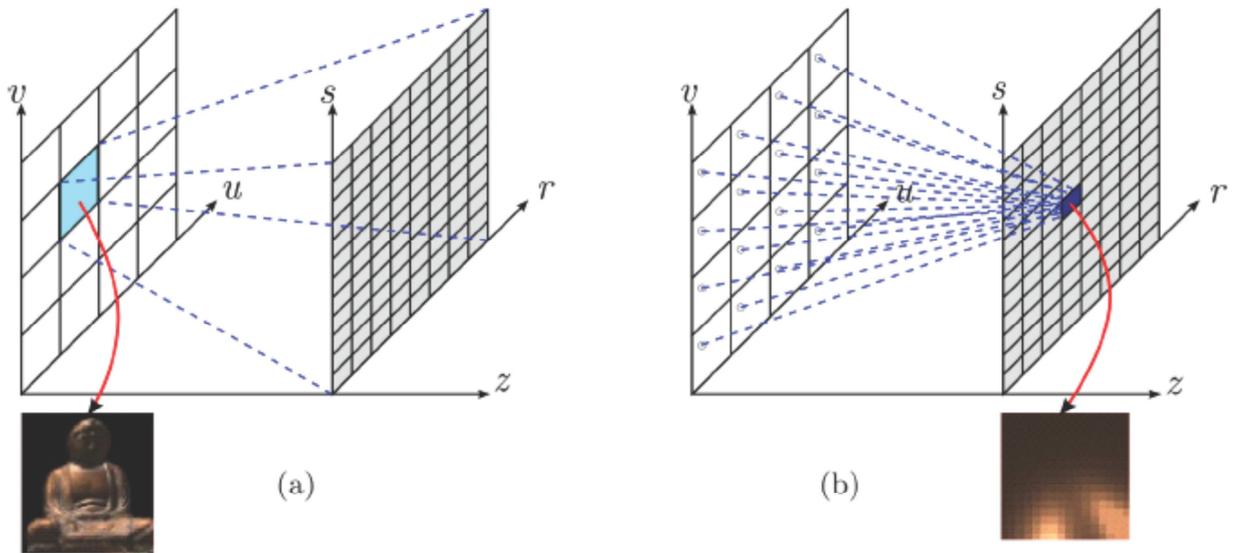


**Figure 4.** *Light Field à 360 degrés.*

Le Lightfield et le Lumigraph ont été présentés au même moment, respectivement par [LH96] et par S. Gortler et al. [GOR96], les deux méthodes sont pratiquement identiques. Elles sont basées sur l'enregistrement des intersections des rayons lumineux de l'environnement avec deux plans. Ces deux plans définissent la boîte englobante de la perception de l'environnement. Ce qui contraint la fonction plénoptique à 4 dimensions,  $(u, v)$  pour le premier plan et  $(r, s)$  pour le deuxième (figure 5).

Le premier plan est défini comme étant le plan de la caméra et le deuxième comme étant le plan focal. Chaque plan est discrétisé en un nombre fini de rayons lumineux à enregistrer. Au moment de l'acquisition, la caméra suit une trajectoire bien définie avec des pas constants. C'est-à-dire tous les rayons lumineux discrétisés qui traversent la surface  $(r, s)$  vont être liés à un point discret du plan  $(u, v)$ . Par conséquent pour chaque point discret du plan  $(u, v)$ , on obtient une image qui correspond à un tableau 2D des rayons optiques. Pour obtenir une nouvelle vue, on commence par la décomposer en éléments discrets sur le plan  $(r, s)$ . Pour un point discret du plan  $(r, s)$  il existe un ensemble de rayons connectés à une zone particulière des imagettes du plan  $(u, v)$ . On utilise cet ensemble de rayons pour calculer par interpolation le point discret du plan  $(r, s)$ . On utilise cette même méthode pour calculer tous les points discrets du plan  $(r, s)$  afin d'obtenir la nouvelle vue. La nouvelle vue est un ré-assemblage des

parties des rayons lumineux enregistrés. On détermine la vue en se plaçant sur un noeud d'intersection de tous les rayons à l'intérieur de l'espace  $(u, v, r, s)$ .



**Figure 5.** Fonctionnement du Light field : (a) principe de l'acquisition de chaque rayon des images dans le plan  $(u, v)$  avec un exemple d'imagette obtenue, (b) principe du rendu d'une nouvelle vue pour un seul élément discret du plan  $(r, s)$  avec un exemple d'échantillon obtenu.

Pour tous les éléments discrets. C'est comme si le plan image glissait par rapport au plan focal. La position de la nouvelle vue de synthèse doit être dans la zone d'espace où les rayons ont été enregistrés, c'est-à-dire l'espace de la boîte englobante. Le rendu d'une nouvelle image demande un temps de calcul important, mais il peut très bien s'effectuer en temps réel [SCG97].

### 3.1.1.4 Panoramas

Quand le point de vue de capture est réduit à un point, la structure de capture de la fonction plénoptique est dénommée Panorama. Ainsi, Chen [Ch95] a construit des images panoramiques cylindriques à 360 degré (à champ de vision vertical limité). Les images panoramiques (figure 6) peuvent être créées par rendu conventionnel, à l'aide d'appareils photos panoramiques spéciaux ou en combinant des photographies entre elles. A chaque point de capture, une rotation et un zoom peuvent être effectués en utilisant une déformation de type cylindre plan. La navigation en 3D se fait de façon discrète, d'un point à l'autre.

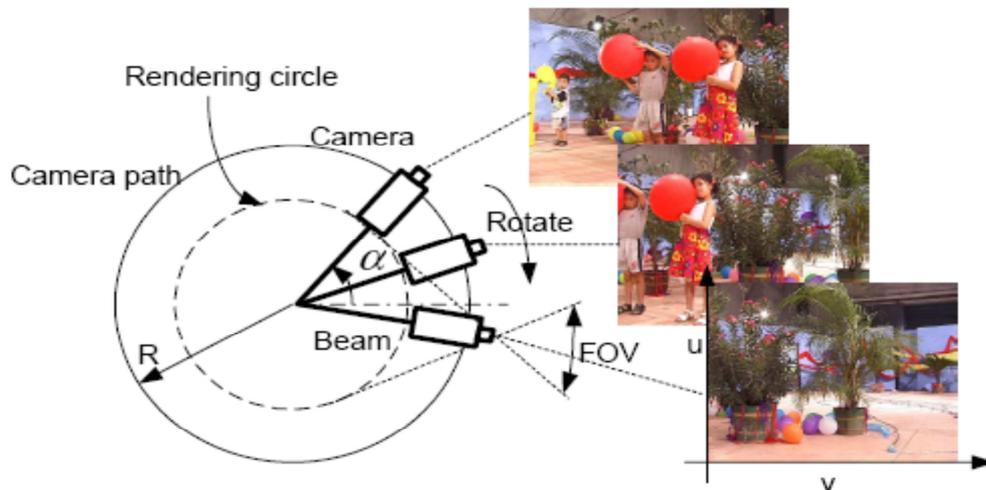


**Figure 6.** *Images panoramiques.*

Les représentations panoramiques peuvent être correctement utilisées dans le cas où le mouvement de la caméra est une rotation pure ou si la scène est plane. Il faut également que les paramètres de caméra soient connus afin de pouvoir assembler les différentes images entre elles. Sinon, la représentation résultante ne sera pas physiquement correcte.

### **3.1.1.5 Mosaïques concentriques**

L'idée des mosaïques concentriques est de contraindre le mouvement de la caméra à des cercles concentriques dans un même plan [SH99]. Cette méthode peut être appliquée au cas de scènes synthétiques et des scènes réelles. Les différentes mosaïques sont construites en assemblant des bandes d'images extraites des images acquises sur un même cercle.



**Figure 7.** *Prise de vues de la mosaïque concentrique.*

Le paramétrage de la représentation est réalisé à partir de trois paramètres qui sont le rayon du cercle de la mosaïque  $R$ , l'angle de rotation  $\alpha$  repérant la position de la caméra l'angle. Les vues virtuelles sont créées en identifiant les rayons lumineux appropriés réalisant l'intersection avec le plan de la vue virtuelle.



**Figure 8.** *Nouvelle scène obtenu par Mosaïques concentriques.*

### **3.1.2 Utilisation de la géométrie de manière implicite**

Les méthodes présentées dans cette partie reposent sur la mise en correspondance d'un petit nombre d'images afin de créer des transitions entre les images en interpolant les positions et les couleurs des pixels dans les images virtuelles. On ne dispose pas de l'information 3D et le problème se situe dans la préservation de la forme 3D des objets afin de rendre les transitions réalistes. Des paires de pixels sont mis en correspondance entre les images de façon à calculer la matrice de transformation permettant de calculer le mouvement apparent des pixels dans l'image interpolée. Les méthodes se différencient par la manière dont les appariements de pixels entre deux images sont réalisés.

#### **3.1.2.1 Interpolation de point de vue**

La première méthode de rendu par géométrie implicite revient à chen [CW93]. Ils proposent une méthode qui permet d'obtenir une vue arbitrairement placée entre les deux vues réelles. Pour cela, il faut déterminer une correspondance dense, c'est-à-dire déterminer la correspondance entre tous les pixels de la première image avec la deuxième. Elle est obtenue par des méthodes de flot optique (optical flow).

La nouvelle vue est calculée par l'interpolation du mouvement estimé des pixels dans chaque image réelle. Cette méthode est très contraignante, il faut obligatoirement que la nouvelle vue soit sur la ligne de base des caméras (ligne virtuelle entre les deux centres optiques des caméras). Une méthode récente élimine cette restriction en donnant plus de liberté à la nouvelle vue [KS05]. Cependant le risque d'ambiguïté dans le rendu de la nouvelle vue oblige les deux caméras réelles à être très proches l'une de l'autre. [Jb07]

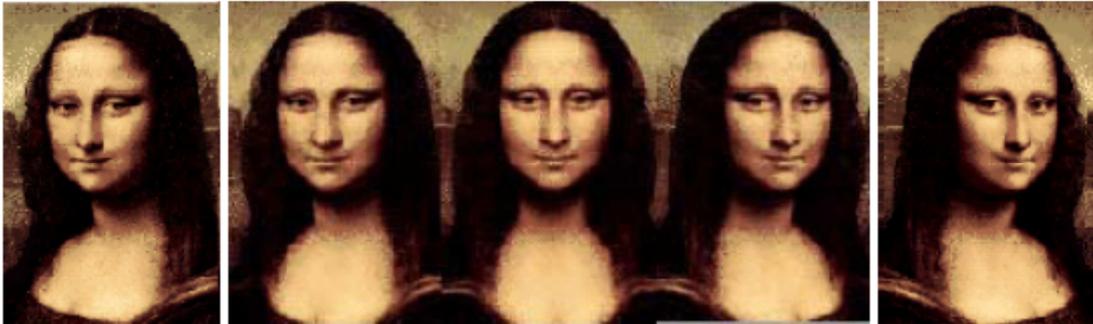


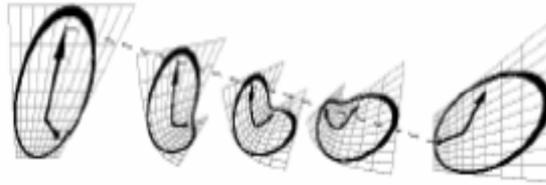
Image de départ

Image d'arrivée

**Figure 9.** *View morphing.*

### **3.1.2.2 Morphing**

Seitz et Dyer [SD97] ont introduit le morphing de vue, qui consiste en une pré-déformation de deux images avant le calcul d'un morphe et ensuite la post-déformation des images interpolées. Ils ont démontré que les techniques de morphing d'image utilisant l'interpolation linéaire pour calculer des positions de particularité dans des images intermédiaires peuvent causer des distorsions 3D sévères. La figure 10 représente une interpolation linéaire de deux vues perspectives d'une horloge (à droit et à gauche) qui cause un effet de courbure géométrique dans les images intermédiaires. La ligne discontinue présente le chemin linéaire d'une particularité pendant la transformation. Cet exemple indique les types des distorsions qui peuvent surgir dans les techniques de morphing d'image [HQU00].



**Figure 10.** *Un morphe déformant une forme.*

### 3.1.3 Utilisation de la géométrie de manière Explicite

Dans cette partie nous décrivons, les techniques où la géométrie de la scène est complètement définie.

#### 3.1.3.1 *Imposteurs*

On peut facilement imaginer que la complexité de la partie visible d'une scène dépasse les capacités matérielles (par exemple, voir une ville complète). Une telle situation oblige, pour que l'on retrouve un temps de rendu interactif, de réduire de beaucoup les petits détails qui, au loin, n'ont plus aucune importance compte tenu de la taille qu'ils occupent à l'écran. Une façon de réduire la complexité consiste à remplacer une partie de la scène par des imposteurs.

##### 3.1.3.1.1 *Les Sprites*

Un sprite est une collection d'images 2D représentant un objet 3D dans diverses positions et sous divers points de vue, les sprites sont très utilisés dans les jeux vidéos 2D, ou bitmaps pour représenter des personnages ou des objets pouvant changer de positions.



**Figure 11.** *Exemple de sprite.*

### **3.1.3.1.2 Les billboards**

Un billboard est un plan texturé placé dans la scène, représentant un objet complexe, Le billboard est un élément essentiel pour décharger le GPU de l'affichage de formes complexes.

Un billboard (en français "panneau") est un plan simulant un objet 3D. Le principe des billboards repose sur le fait qu'à chaque calcul d'image le billboard est positionné pour être face à l'observateur pour cela il peut tourner autour d'un axe ou d'un point de symétrie [Wha05].

L'avantage est de réduire énormément la complexité de la scène, puisque l'on va pouvoir remplacer des objets potentiellement complexes par deux simples triangles texturés formant le plan, la texture bien entendu doit être de qualité.

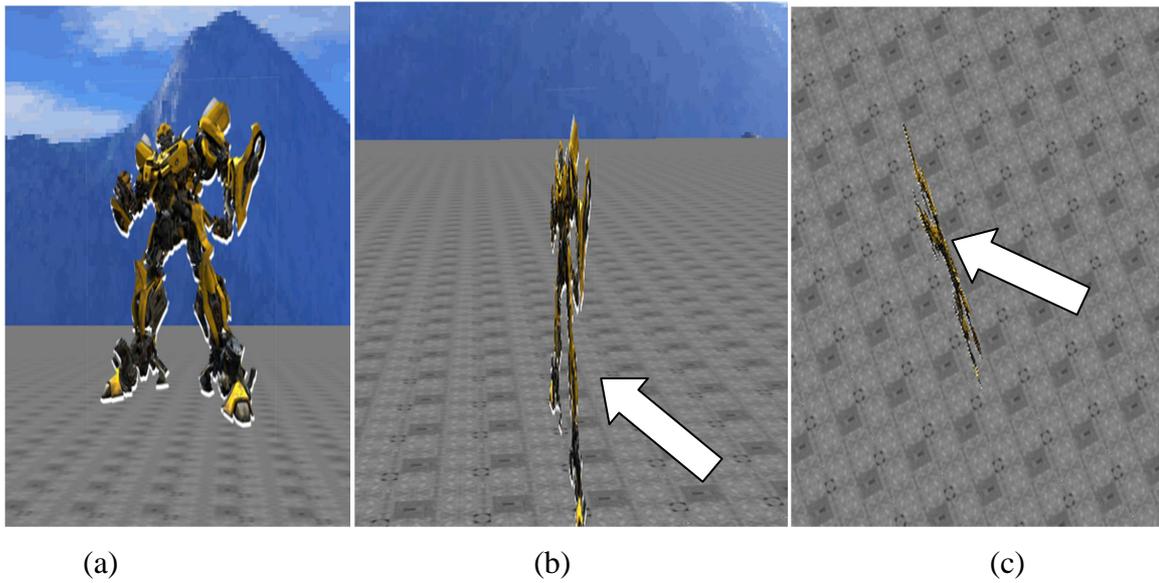
Les billboards sont utilisés pour la végétation, les explosions, les effets météorologiques (nuages, brouillard ...), ou encore des objets très lointains pour lesquels on ne pourra que très difficilement déceler le trucage.

Trois principaux modes d'orientation pour les billboards :

- *Les billboards parallèle à l'écran*

Le principe de ce type de billboard consiste à stocker l'image qui va représenter l'objet complexe dans une texture semi-transparente, puis la plaquer sur un plan, ils ne diffèrent pas beaucoup des sprites, ils sont disposés face au plan image de la caméra. Ils sont efficaces pour représenter un grand nombre d'objets complexes qui possèdent plus de symétrie.

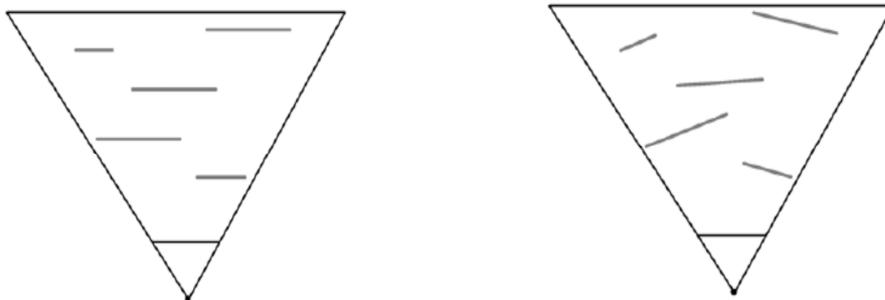
Ces billboards perdent de leur efficacité si la caméra s'en approche trop et ils sont souvent victimes d'un effet « pancarte » qui trahit leur manque d'épaisseur.



**Figure 12.** (a) : billboard face à la caméra, (b) billboard vue de coté, (c) billboard vue dessus.

- Les billboards orientés vers l'observateur

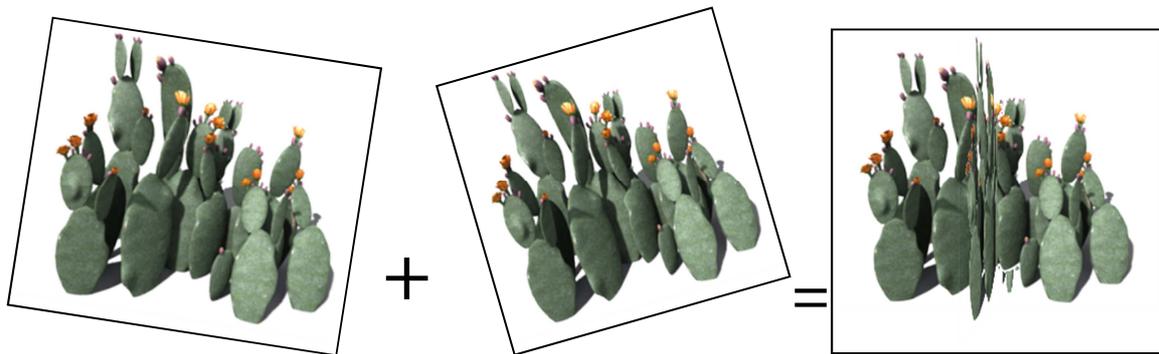
Ils sont un peu plus élaborés que les billboards parallèle à l'écran les billboards sont tracés toujours face à l'observateur. A chaque calcul d'image, le billboard est positionné pour être face à l'observateur, ce modèle représente mieux les objets présentant une symétrie cylindrique. La figure 13 montre la pyramide de projection d'une caméra vue de dessus, les billboards sont orientés vers le centre de la caméra [Pel04].



**Figure 13.** (a) billboards face au plan image de la caméra (b) billboards orientés vers le centre de la caméra.

- *Les billboards croisés*

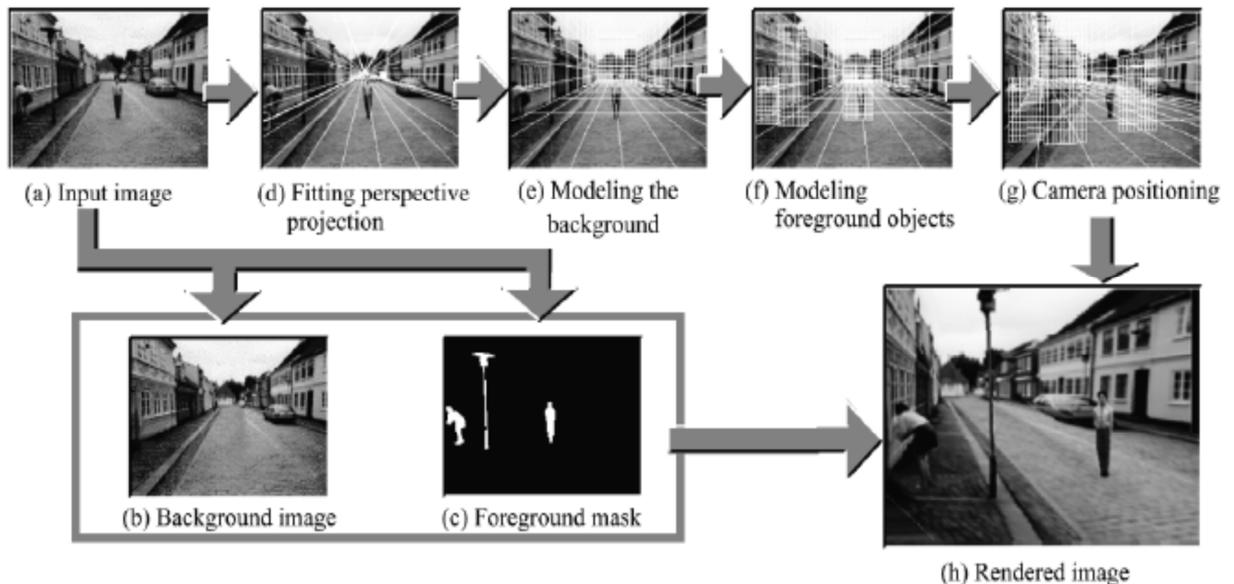
L'utilisation de deux polygones texturés placés en forme de croix pour représenter un arbre, ce qui permet de véritablement tourner autour de l'objet, on peut trouver ce genre de billboards dans des jeux vidéo pour représenter des objets lointains. Lorsque la caméra s'en rapproche, ils sont souvent remplacés par des objets à facettes plus complexes [PC01].



**Figure 14.** *Billboard en croix.*

la méthode « tour into the picture » développée dans [HAA97], Horry et al. utilisent des billboards extraits de photographies dont, le premier traitement consiste à séparer les objets en avant plan de l'arrière plan, pour cela un masque est utilisé pour sélectionner les objets en avant-plan. Une fois ces objets retirés de l'image originale, il reste l'arrière plan et des espaces vides à compléter, pour cela, un logiciel de traitement d'image est adapté en utilisant le pinceau de clonage. Trois images sont obtenues (image originale, masque et arrière plan à compléter) qui servent pour la suite des traitements. A partir de ce moment il faut positionner manuellement les points de fuite dans l'image à l'aide d'un spider mech pour déterminer la projection perspective associée, ensuite une grille peut être déterminée qui permet de déduire des quadrilatères (5 au maximum), des textures doivent être calculer à partir de l'image originale pour être plaquées sur ces quadrilatères lors du rendu, les objet en avant plan sont rendus grâce à des billboards texturés.

Enfin, la caméra est repositionnée de manière à observer la scène d'un autre point de vue. Les billboards sont utilisés pour modéliser les objets aux premiers plans de l'image et sont déplacés lors de l'animation afin d'accompagner le mouvement de pénétration dans l'image



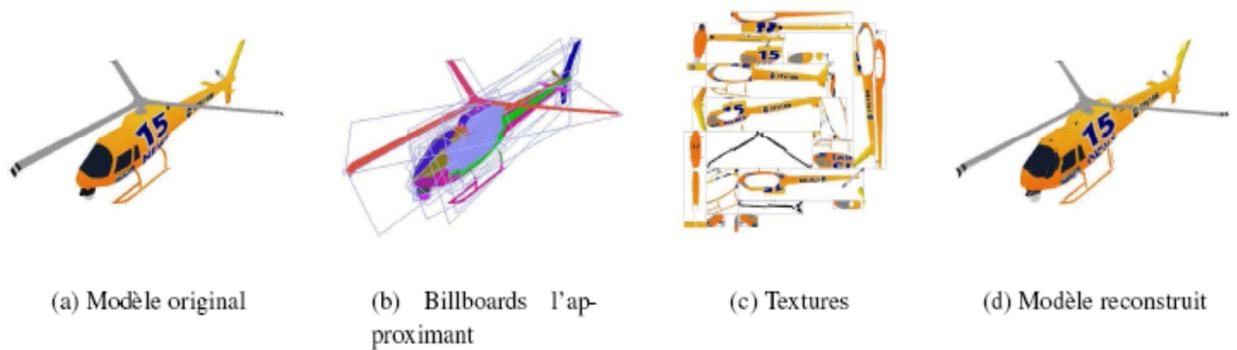
**Figure 15.** *Etapes de "tour into the picture".*

Cette méthode a deux inconvénients majeurs : le premier est l'obligation de régler et de configurer un nombre de Paramètres manuellement et le deuxième est l'ensemble de restrictions concernant la prise de vue originale ( position et orientation ) [HAA97].

### 3.1.3.1.3 Billboard Clouds (nuages de panneaux)

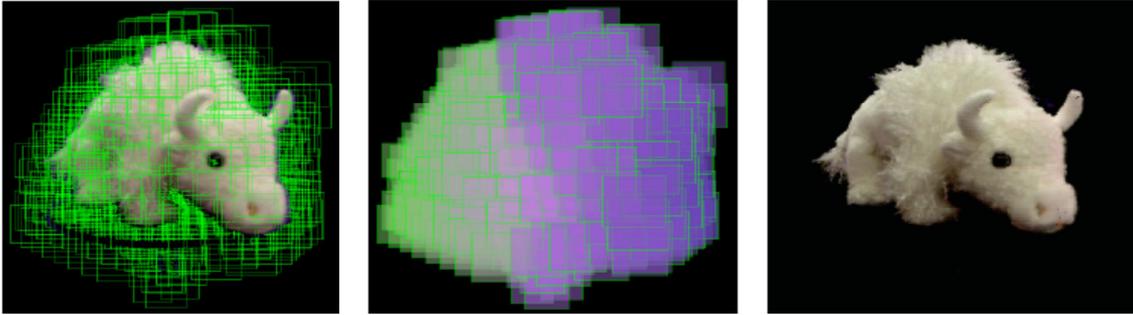
Décoret et al [DDSD03], ont utilisé le principe des billboards pour effectuer l'affichage d'objets géométriques complexes. Le principe de leur méthode est de remplacer l'objet par la combinaison de plusieurs billboards représentant l'objet sous différents angles. Disposés de sorte à approximer le mieux possible la géométrie de l'objet concerné, le résultat est indépendant du point de vue, ce qui permet de ne stocker qu'un seul nuage par objet et donc de ne jamais avoir à effectuer de mise à jour. De plus il est facile de recalculer l'éclairage de l'objet et les algorithmes standards de calcul des ombres par shadow mapping sont directement applicables.

Le principal inconvénient de cette méthode est la lenteur de la phase de construction, qui utilise une transformée de l'espace objet similaire à la transformée de Hough pour calculer les plans approximant le mieux l'objet. De plus les temps de rendu et la consommation en mémoire sont relativement élevés, et certains artefacts peuvent apparaître principalement dans les régions courbes à cause de l'approximation de la géométrie.



**Figure 16.** Exemple billboard cloud.

Yamazaki et al [YSK+02] proposent de générer dynamiquement des billboards à partir d'un ensemble d'images plutôt qu'à partir d'un rendu en images de synthèse. L'objet à modéliser est filmé sur fond bleu avec des caméras calibrées. Une extraction de contours sur chaque image permet alors d'en calculer la géométrie approximative. Un nuage de billboards, ou micro-facettes, est alors disposé de façon cohérente avec cette géométrie. Chacune de ces micro-facettes est orientée vers la caméra et texturée avec les images initiales mélangées de façon judicieuse selon le point de vue de la caméra. Un objet peut être constitué de plus de dix mille micro-facettes, le but étant ici d'avoir un rendu photo-réaliste, notamment sur des objets difficilement modélisables comme des jouets en peluches ou des cheveux. Le rendu s'effectue en temps-réel.



**Figure 17.** *Microfacet billboarding.*

#### **3.1.3.1.4** *Imposteurs statiques*

Maciel et Shirley [MS95] ont introduit les imposteurs pour la visualisation interactive dans un environnement virtuel extrêmement complexe. Dans leurs travaux, la scène complète est modélisée à l'aide d'un maillage géométrique conventionnel et les textures des imposteurs sont stockées dans une structure hiérarchique précalculée appelée "three dimensional image cache".

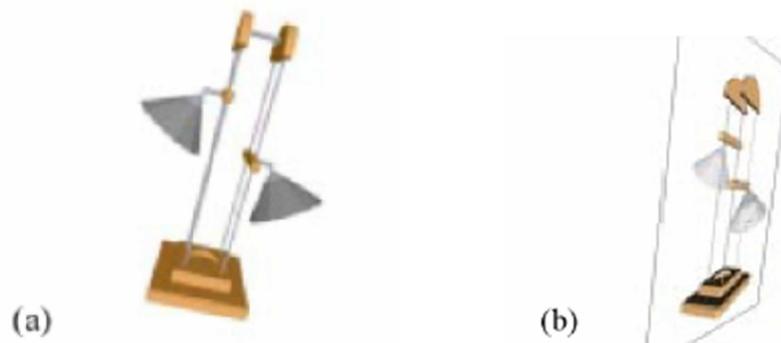
Lors du rendu, la hiérarchie est traversée de haut en bas afin de trouver le meilleur compromis visuel pour le point de vue au pire, quand il y a trop de modèles à afficher en temps réel, une simple boîte englobante texturée est dessinée. Ainsi, le problème principal de cette méthode est de savoir quel imposteur dessiner pour maximiser la qualité visuelle de l'image sans dépasser le temps d'affichage maximum fixé par l'utilisateur [POR04].

L'approche avec des imposteurs précalculés devient rapidement inutilisable dès que le nombre de d'objets augmente car l'espace mémoire nécessaire pour stocker les textures est limité. De plus, la définition des textures ne peut pas être déterminée précisément à l'avance ce qui oblige à toujours prévoir le pire des cas : un imposteur vu de près, nécessitant une haute définition, donc beaucoup de mémoire, alors qu'il est possible que cet objet apparaisse toujours au dernier plan de la scène.

#### **3.1.3.1.5** *Imposteurs dynamiques*

Shafler [Sch95] a introduit les imposteurs dynamiques. Dans cette méthode, un imposteur est toujours représenté par une image de l'objet plaquée sur un polygone transparent Shafler propose donc de générer un imposteur par objet, et ce lors du rendu la scène et ses imposteurs

est stockée dans une structure hiérarchique appelée three dimensional image cache, durant le rendu, les imposteurs sont utilisés pour remplacer les objets complexes. Pour déterminer la validité visuelle d'un imposteur, l'angle maximal en dessous duquel tous les points de la surface de l'objet sont aussi dans l'image est évalué. Tant que l'angle de vue de l'imposteur est inférieur à ce seuil, l'imposteur est considéré valide. Si le point de vue n'est pas translaté mais que seule la direction de vue change, l'imposteur reste valide. Ainsi les imposteurs peuvent être réutilisés d'une image à l'autre lors de l'animation, et ne sont régénérés que si ils deviennent invalides, accélérant ainsi le rendu.



**Figure 18.** *Un objet (a) et son imposteur (b). L'imposteur est vue d'un point de vue différent que la géométrie originale.*

### 3.1.3.1.6 Nailboards et imposteurs en couche

#### **Nailboards**

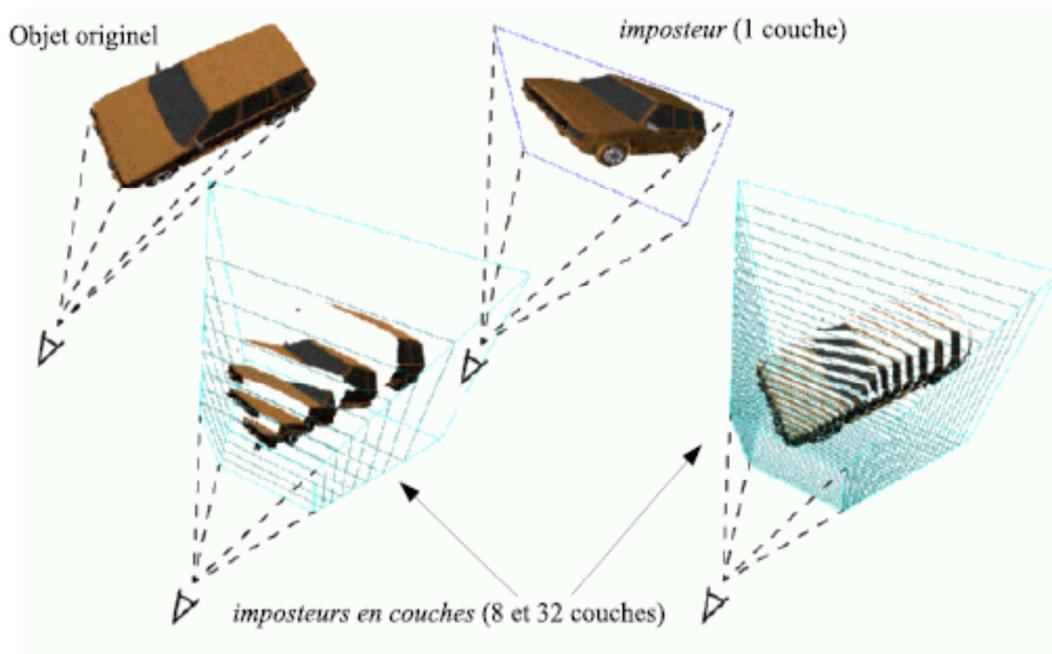
Schaufler [Sch97] introduit « nailboards » Un nailboard est un polygone sur lequel est plaqué une texture de type RVB $\Delta$ , représentant un objet complexe pour chaque texel la composante  $\Delta$  représente la distance entre le polygone de l'imposteur et l'objet réel. Quand cette texture est générée à partir de l'objet réel, la valeur de  $\Delta$  est récupérée directement à partir du tampon de profondeur.

Ensuite, lors du rendu, cette valeur est utilisée pour changer la profondeur du polygone de l'imposteur. Schaufler décrit les transformations nécessaires pour pouvoir intégrer ce nailboard au tampon de profondeur standard et ainsi obtenir une visibilité correcte. Avec cette méthode, un nailboard peut être mélangé au rendu conventionnel qui emmagasine par pixel l'information de profondeur pour calculer la visibilité [POR04].

### ***Imposteurs en couches***

Une extension des imposteurs planaires est celle des imposteurs "par couches" [Sch98]. Il s'agit simplement d'une suite d'imposteurs planaires obtenus en projetant l'objet sur des plans de coupe situés à diverses profondeurs. Les temps de construction et de rendu sont un peu plus élevés que pour les imposteurs planaires, mais restent dans le même ordre de grandeur. Le problème de validité due aux déplacements parallaxes diminue au fur et à mesure que l'on augmente le nombre de couches (pour théoriquement disparaître avec un nombre de couches infini), ce qui fait que la fréquence des mises à jour est sensiblement réduite par rapport aux simples plans.

Un problème de cette représentation est que la quantité de mémoire utilisée devient assez élevée si l'on désire utiliser un nombre conséquent de couches néanmoins, en filtrant les différents plans pour ne conserver que les données congrues, ce phénomène peut être limité. Il reste un défaut majeur de ce type d'imposteur, à savoir qu'il est très propice aux phénomènes de désocclusion : si l'angle de vision de l'imposteur devient élevé, on va voir entre les couches des parties de l'objet qui ne devraient pas l'être, dans le cas où les plans ont été construits de manière optimale [JWS02], on verra également apparaître des trous dans l'objet suite à la désocclusion de zones vides.



**Figure 19.** *Imposteurs en couches.*

### **3.1.3.1.7 Imposteurs maillés**

Les méthodes d'imposteur précédentes plaquent une image sur un plan (quadrilatère, des polygones). Cependant, le nombre d'images pour lesquels ces imposteurs sont valables est très limité en raison des problèmes de parallaxe. Quelques chercheurs utilisent une information géométrique plus sophistiquée.

Ils n'utilisent pas seulement l'image de la géométrie distante comme une texture, mais ils utilisent également un maillage triangulaire pour approximer cette géométrie. Cette méthode, appelée imposteur maillé, triangule la scène en régions approximativement plates dans une étape de prétraitement. Durant l'exécution, la texture est plaquée sur ces triangles [HQU00]. Darsa et al [DCV97] ont présenté une technique de rendu basée sur image pour la navigation dans des environnements tridimensionnels [HQU00, POR04] :

- Un ensemble de cartes d'environnement cubiques est pré-rendu à partir d'un certain nombre de points de vue fixés dans le modèle virtuel.
- Une maille polygonale est créée par triangulation des six cartes de profondeurs associées ainsi obtenues.
- Une vue prise d'un point de vue arbitraire peut être reconstruite par le placage de texture sur les maillages créés.
- Des cartes d'environnement multiples avec leurs mailles triangulaires sont utilisées pour remplir les trous dans les images en combinant les images de profondeur issues de la structure.

Sillion et al [SDB97], ont utilisé la technique d'imposteur pour la visualisation en temps réel de paysages urbains. Le modèle urbain est segmenté en deux sous-ensembles distincts sur chaque frame : Le voisinage local et le paysage distant.

- Le voisinage local est rendu d'une façon normale.
- Le paysage distant est remplacé par des imposteurs maillés.

Un des problèmes avec cette approche est que la séparation imposteur/géométrie réelle est visible lorsqu'on se rapproche du modèle [HQU00]. imposteurs en couches (8 et 32 couches)

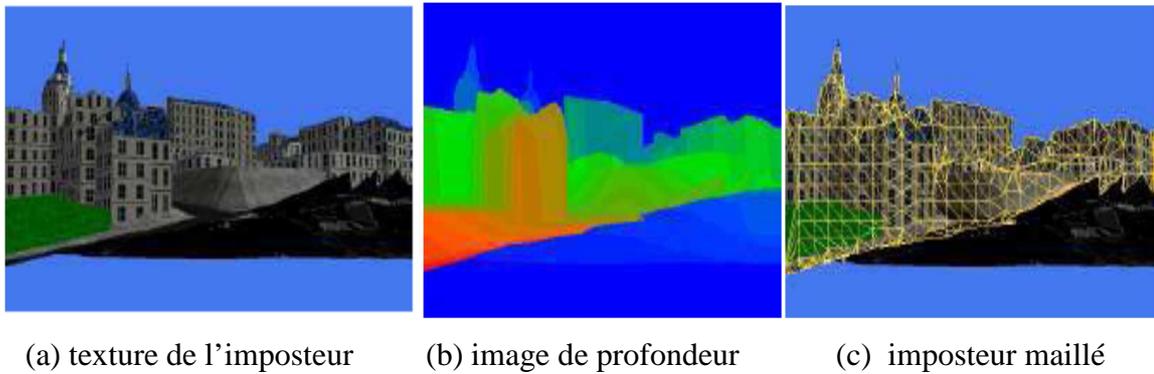


Figure 20. *Imposteurs maillés de Sillion.*

### 3.1.3.2 Déformation 3D d'images

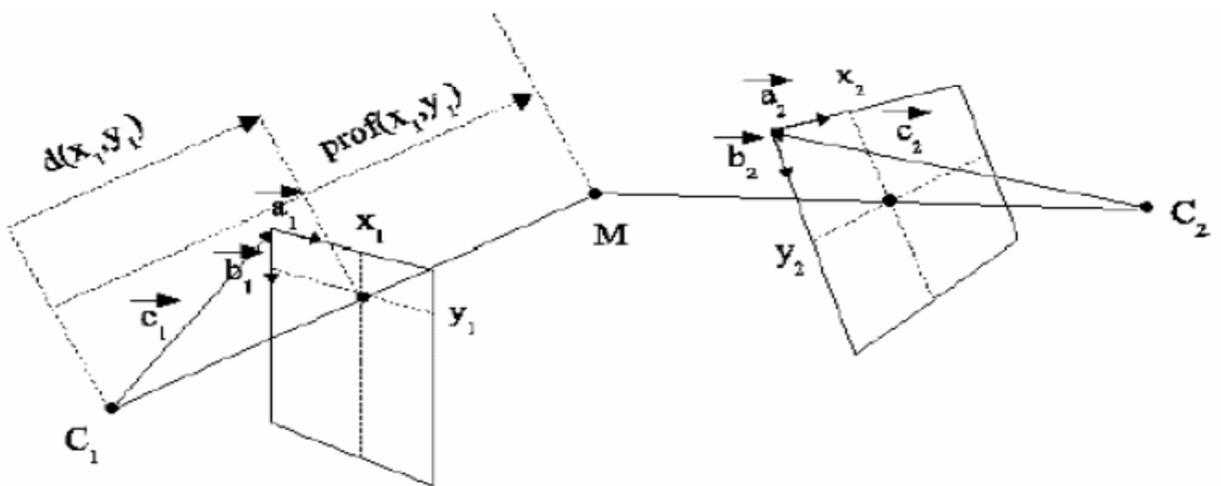


Figure 21. *3D image warping.*

Dans [Mc97], McMillan formalise les équations de reprojection. Il dérive de la profondeur classique la profondeur projective (projective depth) ou encore disparité généralisée (generalized disparity), pour laquelle l'expression des équations de reprojection est plus simple. La profondeur projective d'un pixel correspond au rapport de la distance focale de la

caméra par la profondeur du point La disparité généralisé ( $\delta$ )= rapport de la profondeur du pixel avec la distance pixel-centre de projection :

$$\delta(x_1, y_1) = \text{prof}(x_1, y_1)/d(x_1, y_1)$$

Ainsi, si l'on souhaite calculer le point P2(x2,y2) de I2 à partir du point P1(x1,y1) , de disparité généralisée  $\delta(x_1, y_1)$  on a :

$$x_2 = \frac{\vec{a}_1 \cdot (\vec{b}_2 \times \vec{c}_2)x_1 + \vec{b}_1 \cdot (\vec{b}_2 \times \vec{c}_2)y_1 + c_1 \cdot (\vec{b}_2 \times \vec{c}_2) + C_2 \vec{C}_1 \cdot (\vec{b}_2 \times \vec{c}_2) \delta(x_1, y_1)}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{b}_2)x_1 + \vec{b}_1 \cdot (\vec{a}_2 \times \vec{c}_2)y_1 + c_1 \cdot (\vec{a}_2 \times \vec{c}_2) + C_2 \vec{C}_1 \cdot (\vec{a}_2 \times \vec{c}_2) \delta(x_1, y_1)}$$

$$y_2 = \frac{\vec{a}_1 \cdot (\vec{c}_2 \times \vec{c}_2) x_1 + \vec{b}_1 \cdot (\vec{c}_2 \times \vec{a}_2)y_1 + c_1 \cdot (\vec{c}_2 \times \vec{c}_2) + C_2 \vec{C}_1 \cdot (\vec{c}_2 \times \vec{a}_2) \delta(x_1, y_1)}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{b}_2)x_1 + \vec{b}_1 \cdot (\vec{a}_2 \times \vec{b}_2)y_1 + c_1 \cdot (\vec{a}_2 \times \vec{b}_2) + C_2 \vec{C}_1 \cdot (\vec{a}_2 \times \vec{b}_2) \delta(x_1, y_1)}$$

Le principal inconvénient de ces méthodes vient de l'apparition de trous dans les images synthétisées. Ces trous sont dus à la différence d'échantillonnage entre les surfaces vues des points de vue de référence et des points de vue à créer.

De plus McMillan proposa une méthode permettant de gérer correctement la visibilité lors du rendu, sans utiliser de tampon de profondeur, ainsi qu'une technique de reconstruction d'image incrémentale permettant de remplir l'image de destination par balayage de lignes direct.

### 3.1.3.3 Bilan

Le rendu par champ de lumière permet de mémoriser un grand nombre de rayons lumineux traversant la scène et les effets photométriques qui leur sont associés. Des méthodes de compression ont été proposées afin de rendre ce type de représentation moins coûteux en termes d'espace mémoire. Cette technique possède l'avantage de garder l'aspect photo réaliste du champ de lumière sans passer par une modélisation complexe de l'objet.

Les panoramiques et mosaïques réalisent la mise en correspondance inter-image, il est possible de créer des transitions réalistes entre des paires ou des triplets d'images, par contre,

les occultations éventuelles rendent délicates l'automatisation des différentes étapes qui aboutissent à la mise en correspondance. C'est le cas des techniques de morphing d'images, où les correspondances sont réalisées en général de manière semi-automatique pour avoir des transitions plus réalistes. Nous résumons dans le tableau suivant quelques une des caractéristiques importantes des représentations à champs de lumière.

Dimension	Année	Espace de vision	Méthode
7	1991	libre	fonction plénoptique
5	1995	libre	modélisation plénoptique
4	1996	plan	Le Lightfield et le Lumigraph
3	1999	Cercle	Mosaïques concentriques
2	1994	Point fixe	Panoramas

**Tableau 2.** *Caractéristiques des techniques de champ de lumière.*

Une façon de réduire la complexité consiste à remplacer une partie de la scène par des imposteurs.

Cette méthode consiste à remplacer une géométrie complexe par des polygones texturés, plus simples à afficher, mais offrant un résultat visuel comparable.

Cependant, il existe encore plusieurs problèmes pour cette technique. Les panneaux texturés demeurent encore relativement compliqués à construire et sont encore plus difficiles à gérer correctement, d'autres techniques remplaçant la géométrie éloignée par une géométrie simplifiée existent les imposteurs offrent une manière commode pour avoir le temps nécessaire pour rendre une partie de la scène, dépendant du nombre de Pixel qu'elle couvre sur l'écran plutôt que sur la complexité de sa représentation géométrique.

Cependant, il faut être attentif au gros besoin en espace mémoire pour stocker les imposteurs, le tableau suivant montre un comparatif avantages /inconvénient des techniques d'imposteurs :

La technique	Les avantages	Les limites
Les imposteurs statiques	<ul style="list-style-type: none"> <li>- simple à mettre en œuvre et très efficace</li> </ul>	<ul style="list-style-type: none"> <li>- consommation élevée en mémoire.</li> <li>- des artefacts visuels peuvent apparaître</li> <li>- l'impossibilité de changer l'illumination ou l'ombrage.</li> </ul>
Les imposteurs dynamiques	<ul style="list-style-type: none"> <li>- les imposteurs sont générés durant la phase de rendu.</li> <li>- Les imposteurs peuvent être réutilisés d'une image à l'autre lors de l'animation.</li> </ul>	<ul style="list-style-type: none"> <li>- Les mises à jour fréquentes de l'imposteur dynamique</li> <li>- augmente le temps de d'exécution</li> </ul>
Les imposteurs en couche	<ul style="list-style-type: none"> <li>- Augmentation du nombre des couches permettent de résoudre parfaitement les problèmes de visibilité.</li> </ul>	<ul style="list-style-type: none"> <li>- Le phénomène de désocclusion (on verra apparaître des trous dans l'objet).</li> <li>- Consommation élevée en mémoire.</li> </ul>

Billboard Clouds	<ul style="list-style-type: none"> <li>- aucune mise à jour à effectué.</li> <li>- l'éclairage et l'ombrage facilement calculable.</li> </ul>	<ul style="list-style-type: none"> <li>- la phase de construction est très lente.</li> <li>- le temps de rendu et la consommation en mémoire sont relativement élevés.</li> <li>- des artefacts peuvent apparaître.</li> </ul>
Microfacettes billboard	<ul style="list-style-type: none"> <li>- le rendu final est très rapide.</li> <li>- un grand réalisme.</li> </ul>	<ul style="list-style-type: none"> <li>- l'éclairage ne peut varier du fait de l'emploi de placage de texture dépendant du point de vue.</li> </ul>

**Tableau 3.** *Comparaison des techniques d'imposteurs .*

## 3.2 Représentation des scènes dynamiques

Dans cette partie nous décrivons les représentations qui nous permettent de traiter les scènes dynamiques (variable dans le temps) au lieu des images immobiles, ces méthodes effectuent des prétraitements sur les séquences vidéos, ces prétraitements sont stockés puis utilisés pendant le visionnage des vidéos pour recréer de nouvelles vues, permettant ainsi de choisir son point de vue de façon interactive. La majorité de ces techniques nécessitent toutefois l'utilisation de matériel puissant.

### 3.2.1.1 *Virtualized Reality (Réalité virtualisée)*

Pour obtenir des informations 3D complètes sur la scène, celle-ci doit être observée depuis plusieurs points de vue simultanément. Un tel système multi-caméra ne peut cependant pas être construit par simple juxtaposition de systèmes mono-caméra. En effet des problèmes spécifiques apparaissent : acquisition et traitement synchrone des images, gestion du flux

important des informations brutes (images couleurs), extraction des informations 3D coûteuse.

La plupart des systèmes multi-caméras se sont intéressés à l'extraction de la géométrie de l'utilisateur. Cette information tridimensionnelle est très riche car elle permet d'obtenir le positionnement global de l'utilisateur.

Le premier système multi-caméra Virtualized Reality fut proposé par Kanade et al. [KNR95], et développée dans [KRN97, RNK97]. Il est composé d'un dôme comportant 49 caméras, synchronisées et calibrées. L'acquisition des images est réalisée dans un premier temps sur plusieurs disques de manière synchrone grâce à un système électronique. Pour chacune de ces images, une carte de profondeur est calculée par une MBS [OK93] (Multi-baseline Stereo).



**Figure 22.** *Réalité virtualisée.*

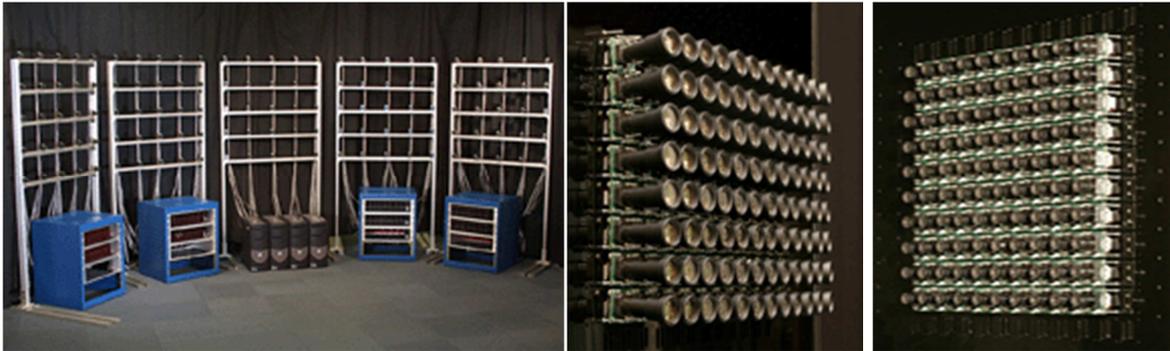
Pour générer une nouvelle image à partir d'un point de vue donné, une reconstruction de référence est choisie. Cette reconstruction est issue de la carte de profondeur de la caméra la plus proche du point de vue traité. Ce choix permet de minimiser le nombre de trous dans l'image générée. Pour améliorer le rendu, les trous sont détectés sur l'image générée puis éventuellement comblés par des reconstructions issues de caméras voisines. Le rendu supplémentaire n'est évidemment effectué que sur les trous. Il se peut malgré tout que les trous ne soient pas totalement comblés après cette étape.

Virtualized Reality est donc une méthode permettant de générer de nouvelles images d'une scène dynamique en différé. Cette méthode nécessite un nombre conséquent de caméras (une cinquantaine) et un dispositif particulier comme par exemple un dôme pour disposer les caméras de façon adéquate.

Ce système n'est pas interactif mais hors-ligne : aucune contrainte de performances n'est donc imposée [Vn06].

### 3.2.1.2 Murs de caméras

Un murs de caméras est un dispositif constitué d'un grand nombre de caméras, le principal défi de ce genre de dispositif est la façon de gérer l'énorme flux de données transitant des caméras vers la ou les machines de traitement. Ceci explique pourquoi les premiers murs de caméras se limitaient à une cinquantaine de caméras.



**Figure 23.** *Mur de caméras .*

Le plus connu des murs de caméras est probablement le Stanford Camera Array (figure 23), il peut supporter plus d'une centaine de caméras, Les caméras utilisées et notamment leur focale varient selon que la scène soit intérieure ou extérieure, la résolution reste la même : 640×480. Chaque caméra est connectée à un FPGA (carte programmable) qui se charge de générer des fichiers RAW ou de la compression en MPEG2. Le FPGA sert aussi de routeur afin de transmettre les signaux de synchronisation aux caméras voisines. Les FPGA sont reliés entre eux sous forme d'arbres binaires et la caméra racine est connectée au PC qui les contrôle. Si le nombre de caméras devient trop élevé, il suffit d'ajouter un PC supplémentaire qui gèrera son propre groupe de caméras. Ainsi, une centaine de caméras nécessitent 4 PC pouvant gérer un débit de 30 fps. Pour plus de détails techniques sur ce dispositif, le lecteur peut se référer à Wilburn et al. [WSK02].

### 3.2.1.3 Bilan

Parmi les méthodes détaillées dans cette section, la méthode de Virtualized Reality a été la première proposée et a eu un impact assez fort, même si la qualité visuelle des images générées n'était pas excellente.

Le mur de caméras de Stanford est lui aussi très impressionnant, tant du point de vue de la quantité de matériel mis en œuvre qu'au point de vue des résultats obtenus. L'avantage de cette méthode est d'être performante sur d'autres applications que le rendu à base de vidéos, c'est le cas en particulier pour la rectification dynamique des couleurs et pour la simulation de profondeur de champ.

## 3.3 Techniques hybrides

### 3.3.1 Le placage de texture

Le placage de texture est le modèle d'habillage le plus répandu. Il est utilisé dans les simulateurs, les jeux vidéo, mais également dans les logiciels de rendu réaliste. Il est aujourd'hui un composant de base de la synthèse d'image.

Le placage de texture a été introduit par Catmull [Cat74], L'idée principale est d'associer une image à deux dimensions appelée texture à une surface géométrique. L'apparence de la texture est définie dans un espace de référence : l'espace texture. La correspondance entre l'objet et la texture est définie par la paramétrisation de la surface géométrique. On associe chaque point de la surface à un point de l'espace texture, appelé coordonnée de texture ( $u, v$ ). La paramétrisation permet de lire directement dans la texture la couleur correspondant à un point quelconque de la surface. La texture est ainsi appliquée à la surface de l'objet, comme illustré la **Figure 24**.



**Figure 24.** *Placage de texture.*

Elle présente plusieurs limitations, En effet, le placage de texture ajoute des détails à une surface en modifiant la couleur, mais cette couleur reste la même quelque soient les conditions d'éclairage, ce qui révèle que le relief représenté par l'image n'est qu'une illusion. En plus, le placage de texture ne fonctionne bien que pour des surfaces planes ou légèrement courbées et les objets réalistes très détaillés nécessitent beaucoup de textures, qui peuvent de plus nécessiter d'être traitées en plusieurs passes, ralentissant d'autant les performances. Des phénomènes tels que la fumée, le feu ou l'eau sont difficiles à traiter en employant cette méthode.

Plusieurs variantes et extensions de cette technique ont été proposées pour remédier à ces limitations et augmenter le réalisme des résultats comme par exemple, le placage de bosselures (BumpMapping), l'ombrage par carte d'horizon (Horizon Mapping) et le Déplacement de Surface (Displacement Mapping). Mais l'extension la plus remarquable est celle proposée par M. Oliveira [OBM00]: le placage de texture en relief (Relief texture mapping).

### **3.3.2 Placage de texture en relief**

Le placage de texture en relief proposée par M.Oliveira [OBM00] est une extension de placage de texture conventionnel qui permet la représentation 3D de détails sur les surfaces et génère des effets de parallaxe.

Cette technique produit des vues correctes des surfaces par la déformation de texture augmentées par des informations de profondeur en chaque pixel.

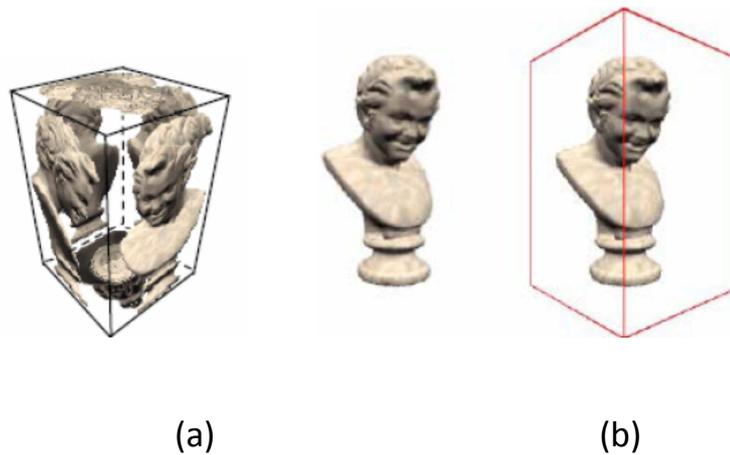
L'opération de placage de texture en relief s'opère en deux passes, durant la première passe de pré-déformation (pre-warping), la texture est déformée en fonction de la direction d'observation. Cette phase est lente et effectuée par le CPU. La seconde passe est très rapide et ne consiste qu'à plaquer la texture déformée de manière ordinaire. Cette passe est généralement exécutée par la carte graphique.

Il s'avère que les textures en relief offrent des possibilités intéressantes, elles augmentent de manière significative le réalisme visuel de la scène rendue sans aucune charge supplémentaire sur le système de rendu.



**Figure 25.** Exemples de reliefs produits par Oliveira [OBM00].

La texture en relief peut être utilisée aussi pour la reconstruction et la visualisation d'objets 3D à partir d'un ensemble de vues entourant l'objet considéré.



**Figure 26.** (a) Objet représenté par six textures en relief, (b) vue de l'objet reconstruit par placage de texture en relief.

## 4 Conclusion

Nous avons vu, tout au long de ce chapitre, que les différentes classes de méthodes que nous avons présentées ont toutes pour but d'obtenir une grande complexité visuelle à moindre coût possible en terme de temps d'affichage et d'espace mémoire.

Les représentations basées images décrivent la scène directement à partir des images acquises dans la scène. Le principal inconvénient des approches basées images est la quantité de données requise pour recréer l'impression d'immersion dans un environnement complexe. On aboutit alors à la notion de compromis entre la compacité de la représentation et l'interactivité autorisée avec la scène. La recherche d'une représentation compacte passe par la complexité des liaisons qu'il est possible de créer entre les différentes images.

Ainsi, sans aucune liaison entre images, on trouve les représentations comme les champs de lumière qui ont besoin d'une base de données très grande pour assurer une bonne qualité d'image. En effet, une représentation par champ de lumière permet de mémoriser un grand nombre de rayons lumineux traversant la scène, et les effets photométriques qui leur sont associés.

Les techniques de rendu par imposteurs permettent une vitesse de rendu élevée au prix d'une liberté de mouvements réduite, cependant elles s'en différencient par une plus grande liberté offerte au prix d'un rendu moins rapide : celui-ci ne sera plus indépendant de la complexité de la scène, et sera essentiellement lié au nombre d'objets composant la scène, le temps de rendu d'un imposteur donné sera quant à lui généralement assez peu dépendant de la complexité de l'objet.

Les images étant générées séparément pour chaque objet, il va être possible de faire évoluer la scène en déplaçant les objets au sein de celle-ci, mais aussi en faisant varier l'éclairage pour peu que l'on ait conservé en plus des informations sur la couleur des objets et les données sur ses normales. Cependant l'animation d'un objet lui-même reste problématique et obligera sans doute à un recalcul complet de son imposteur.

Les imposteurs sont idéalement adaptés à la représentation d'objets plats ou composés de plans, mais offrent une assez médiocre représentation aux objets possédant des formes courbes, à moins bien sûr que ces courbes soient situées uniquement dans des plans (on peut prendre l'exemple trivial d'un disque).

Les méthodes de rendu à base de vidéos utilisent soit un grand nombre de caméras, soit un nombre de caméras plus réduit mais à grande résolution. Un des problèmes majeur consiste alors à gérer le flux vidéo en direct, mais il se trouve que toutes ces méthodes nécessitent un prétraitement sur les images.

Ce prétraitement est long et ne peut être effectué en temps réel. Notons aussi qu'un prétraitement sur des vidéos implique qu'il est nécessaire de les stocker ce qui requiert une quantité de mémoire non négligeable.

## **Chapitre II :**

# **REPRESENTATION DES FORETS EN SYNTHÈSE D'IMAGE**

---

# REPRESENTATION DES FORETS EN SYNTHÈSE D'IMAGE

---

## 1 Introduction

Rendre des scènes naturelles de manière réaliste est l'un des grands problèmes en synthèse d'image, généralement elles peuvent contenir des phénomènes et des objets naturels, on distingue phénomène et objet de la façon suivante :

**Un objet naturel** : solide et concret il, interagit avec la lumière selon les modèles de réflexion matière lumière dans le cadre de synthèse de paysage, ce sont les terrains et les végétaux ....

**Un phénomène naturel** : c'est un élément qui n'a pas de géométrie, ni de matière mais il agit sur la lumière, parmi ces phénomènes, on peut citer l'eau, le brouillard et tous les phénomènes atmosphériques [Cc97].

Dans ce chapitre nous nous intéressons à un élément essentiel des objets naturels qui est les forêts. Rendre de manière réaliste une forêt est une tâche difficile, car on peut remarquer que celle-ci est constituée d'une multitude d'arbres, tous différents les uns des autres, eux même construits selon des lois naturelles relativement complexes

Cette diversité produit une complexité à tous les échelles, une forêt peut compter plusieurs milliers d'arbres de type différent et un arbre peut contenir plusieurs types de feuilles, ce qui produit plusieurs milliards de primitives. Cette complexité est à l'origine de deux problèmes majeurs liés à la production d'arbres en synthèse d'images : la lourdeur du traitement et la résolution trop fine des détails.

La complexité à toutes les échelles de ces structures naturelles pose donc un défi pour les techniques traditionnelles de modélisation et de rendu des scènes 3D.

Nous étudions dans ce chapitre quelques méthodes pour représenter des arbres en images de synthèse.

## 2 Représentation d'arbre en synthèse d'image

Comme l'a décrit Chaudy [Cc97], Il existe deux types de représentations informatiques des arbres :

- **la représentation structurelle :**

qui construit la structure détaillée de l'arbre pour le visualiser ultérieurement à l'aide d'une technique de rendu quelconque.

- **la représentation impressionniste :**

qui met l'accent sur l'impact visuel de l'objet en général, au détriment de sa complexité structurale, le but est d'obtenir un résultat visuel qui ressemble de façon satisfaisante à un arbre lorsqu'on l'observe d'assez loin, et qui est généralement utilisée dans les applications « temps réel ».

### 2.1 Les modèles structureles

#### 2.1.1 *Les premiers modèles à caractère géométrique*

Honda [Hon71] et Aono et Kunii [AK84] ainsi que Kawaguchi [Kaw82] sont les premiers à s'intéresser à la modélisation informatique des végétaux, leurs approches sont purement géométrique. Elle repose pour la génération de structure d'arbre sur l'utilisation d'un nombre important de paramètres géométriques comme la taille du fût de l'arbre, les dimensions des branches, leurs angles, le nombre de troncs de cônes composant chaque branche, le nombre de sous-branches, le nombre de feuilles, le niveau de récursivité.

Aono et Kunii [AK84] proposent de nouveaux modèles géométriques pour calculer la direction et les caractéristiques des ramifications. Les paramètres des formules mathématiques se 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'influence externe 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.

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

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

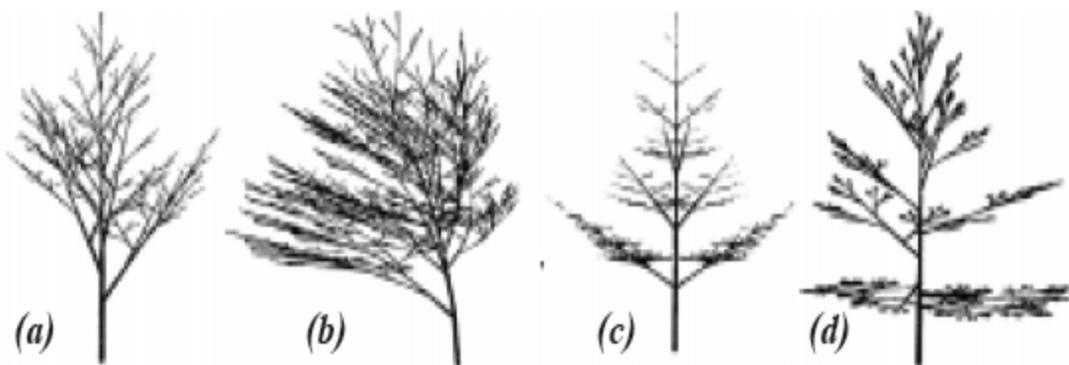
$P$  : est la position de la branche mère.

$P_i$  : représente la position de  $i$  branches fille.

$R_i$  : le facteur de contraction de la branche.

$h_i$  : les angles des 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.



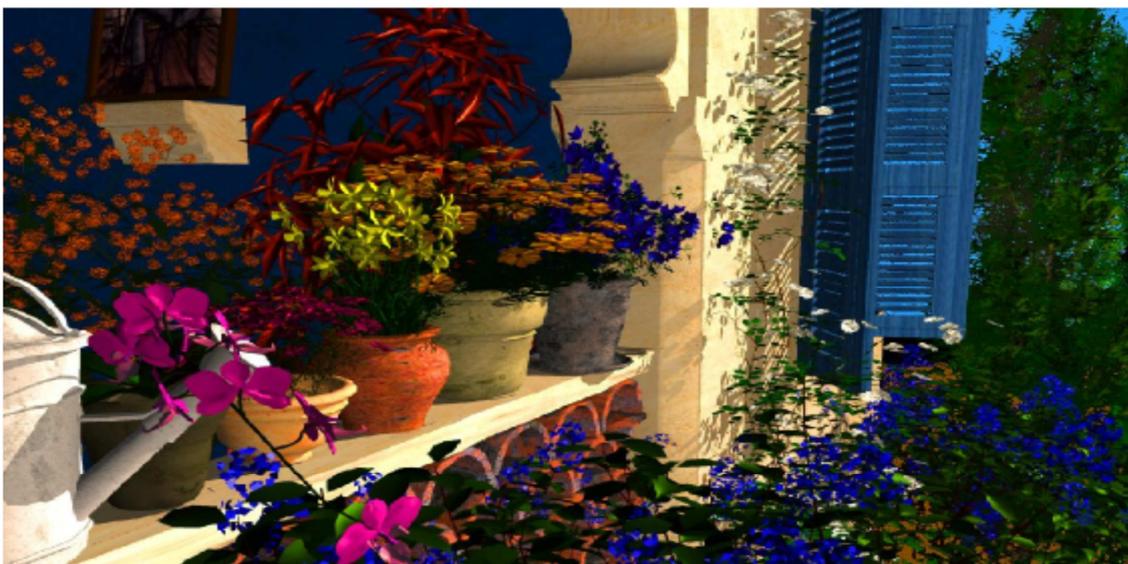
**Figure 27.** Les apports des différents modèles de Aono : (a) modèle de base GMT1, (b) modèle GMT2 avec déviation, (c) modèle GMT3 avec ramification ternaire, (d) modèle GMT4 avec variation des paramètres de branchement en fonction du niveau de croissance [AK84].

### 2.1.2 Les modèles à base de paramètres botaniques

De Reyffye [DREFa88] proposent en 1988 un modèle botanique pour la synthèse de végétaux. Cette approche “biologique” de la modélisation des plantes a pour principe la simulation de l’activité des bourgeons sur une échelle de temps discrète. Des probabilités, caractéristiques de l’espèce et du niveau de développement du végétal considéré, sont affectées à chacun des nœuds d’évolutions possibles du bourgeon. Ces données sont issues des connaissances en botanique et en morphologie des végétaux. Des facteurs externes peuvent également être intégrés, comme par exemple, la coupe de certaines branches ou des maladies, l’effet de la gravité sur les branches, la présence d’un obstacle entre la lumière et une partie de la plante, etc.

La description issue du moteur de croissance est interprétée comme un ensemble de primitives géométriques simples (troncs de cônes pour les branches et polygones pour les feuilles) qui peuvent ensuite être traitées comme n’importe quel autre objet de synthèse d’images, ce qui facilite leur utilisation par un modeleur pour leur intégration à un paysage, ainsi que par un moteur de rendu [MEY01].

Ce modèle très complet et réaliste a abouti à un système de simulation développé par le CIRAD (Centre de coopération Internationale en Recherche Agronomique pour le Développement).

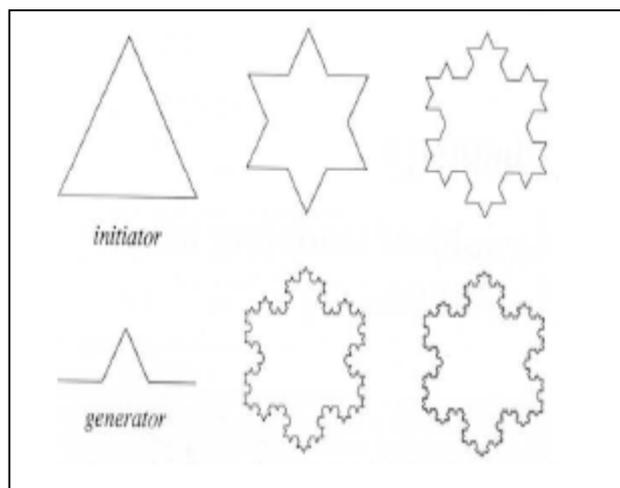


**Figure 28.** Exemples de plantes générées par un modèle à base de paramètres botaniques.

### 2.1.3 Les modèles à base de grammaires

Un L-Système ou Lindenmayer system est une grammaire formelle (grammaires de réécriture parallèle) permettant de modéliser le processus de croissance des plantes. Les L-systèmes ont été introduits et développés en 1968 par le biologiste et botaniste Aristid Lindemayer [Lin68], basée sur une forme récursive de grammaire générative, cette grammaire a été approfondie et mise en œuvre graphiquement par Przemyslaw Prusinkiewicz [PL90].

Le concept central des L-systèmes est un mécanisme de réécriture parallèle de chaînes de caractères basé sur un jeu de règles de production. En général, la réécriture est utilisée pour définir des objets complexes par remplacement successif de parties d'un objet initial simple. L'exemple classique est La courbe du flocon de neige, proposée par Von Koch et illustrée sur la figure 29.



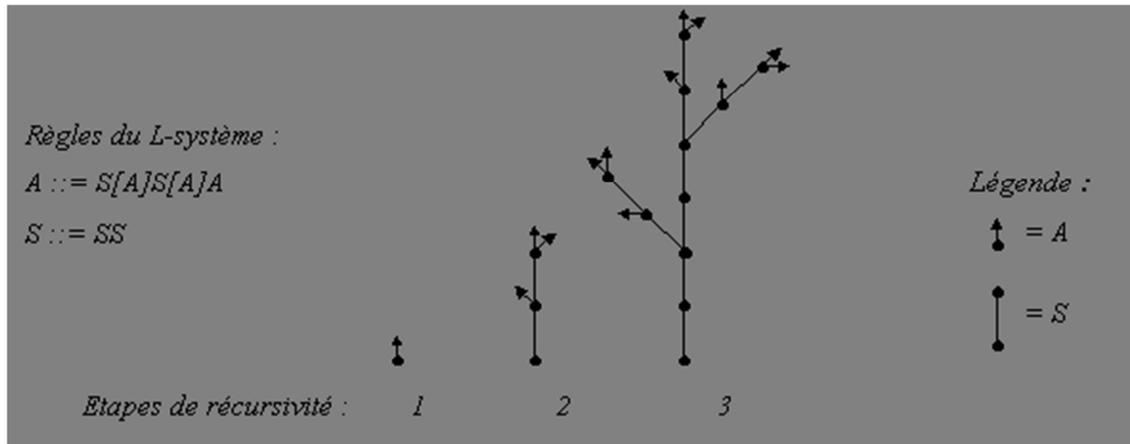
**Figure 29.** Construction du flocon de neige. A chaque itération, les segments sont remplacés par le générateur.

Pour définir des structures à branches telles que les arbres axiaux, Lindenmayer définit une représentation sous forme d'arbre : les chaînes parenthésées. Chaque expression parenthésée correspond à une branche (un sous-arbre) de l'arbre courant.

Deux nouveaux symboles sont alors définis. Ils sont assimilables à des opérations d'empilement et de dépilement de l'état du module courant :

[ : Empiler l'état courant.

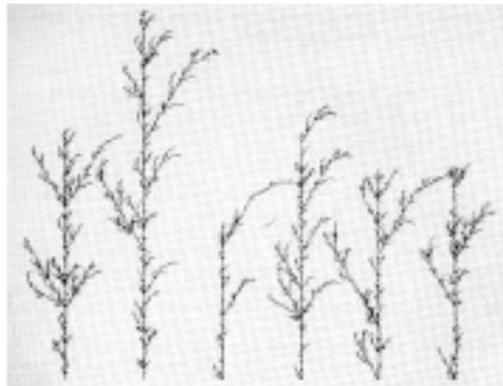
] : Dépiler l'état courant.



**Figure 30.** Exemple simple de L-système dont l'axiome est  $A$ . On peut assimiler  $A$  à un bourgeon et  $S$  à un segment d'entre-noeud.

Mais la combinaison de ces arbres (entièrement déterministes et identiques) dans une image produirait des effets d'une régularité artificielle.

Afin de produire plusieurs spécimens différents d'une même espèce de plantes, Prusinkiewicz introduit un facteur aléatoire dans les L-système, ajoutant une probabilité à chaque règle de production.



**Figure 31.** Structures stochastiques avec branchements.

La modélisation géométrique des topologies générées se fait simplement en interprétant les symboles de la chaîne par un système de type tortue graphique .

Les principaux symboles utilisés avec la géométrie de la tortue sont :  $+$  et  $-$ ,  $<$  et  $>$  qui procèdent à une rotation positive ou négative à la direction courante, et les crochets  $[$  et  $]$  permettent d'empiler et de dépiler l'état de la tortue à un instant donné, ce qui permet de partir sur la création d'une branche puis de revenir à sa racine pour continuer le tracé.

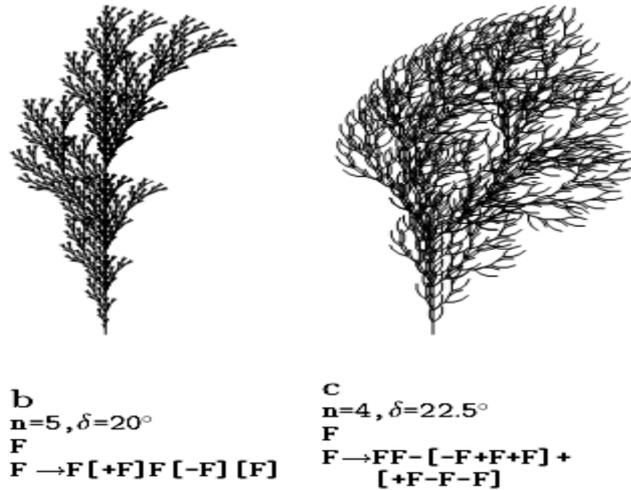


Figure 32. Deux L-systèmes construits par une application récursive de la même règle.

### 2.1.4 Les modèles combinatoires

Viennot et al. [VEJA89] utilisent ce qu'ils appellent une « matrice de ramification ». Celle-ci permet de déterminer la probabilité pour qu'un segment de branche d'ordre  $k$  se décompose en deux sous-branches d'ordres  $j$  et  $k$  avec  $j < k$  ou bien toutes les deux d'ordre  $k-1$ , sachant que l'ordre des segments terminaux est 1. En d'autres termes, cela permet de représenter des grosses branches à partir desquelles partent de plus petites branches ou qui prennent fin pour donner naissance à de plus petites branches (cela peut être le cas du tronc par exemple). Cette matrice est triangulaire et la première ligne est facultative car les noeuds d'ordre 1 sont terminaux et ne peuvent donc pas se décomposer. De plus, la somme des probabilités de chaque ligne de la matrice doit être égale à 1. La figure 33 montre un exemple de représentation d'arbre par cette méthode.

Exemple :

$$\begin{bmatrix} 3/5 & 2/5 \\ 1/2 & 0 & 1/2 \end{bmatrix}$$

peut donner l'arbre

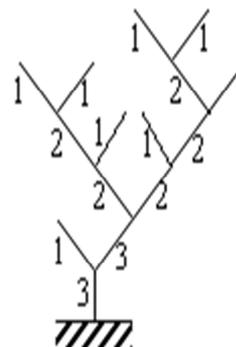


Figure 33. Exemple de représentation d'arbre avec une matrice de ramification.

Dans cet exemple, la première ligne (d'ordre 2) indique qu'un segment d'ordre 2 possède trois chances sur cinq de se poursuivre en générant un segment d'ordre 1 et a deux chances sur cinq de se décomposer en deux segments d'ordre inférieur (donc d'ordre 1). Pour la seconde ligne de cet exemple, on constate qu'un segment d'ordre 3 ne peut pas se poursuivre tout en générant un segment d'ordre 2 (probabilité nulle).



**Figure 34.** Deux arbres générés par la méthode des matrices de ramification [VEJA89].

## 2.2 Les modèles impressionnistes

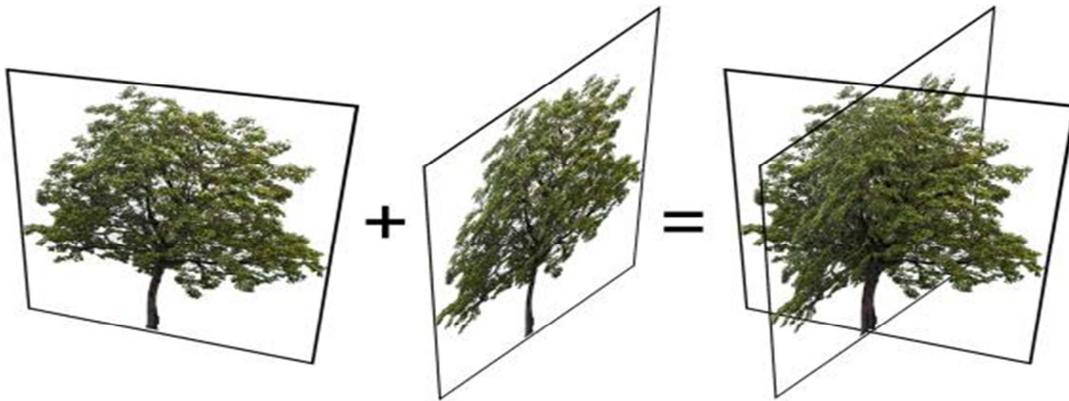
### 2.2.1 Les modèles à base de textures

Une technique simple pour visualiser des arbres est utilisée dans les systèmes de rendu temps réel (simulateurs, jeux vidéo). 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 visualiser les systèmes de rendu temps réel, ces systèmes 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.

Enfin 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.



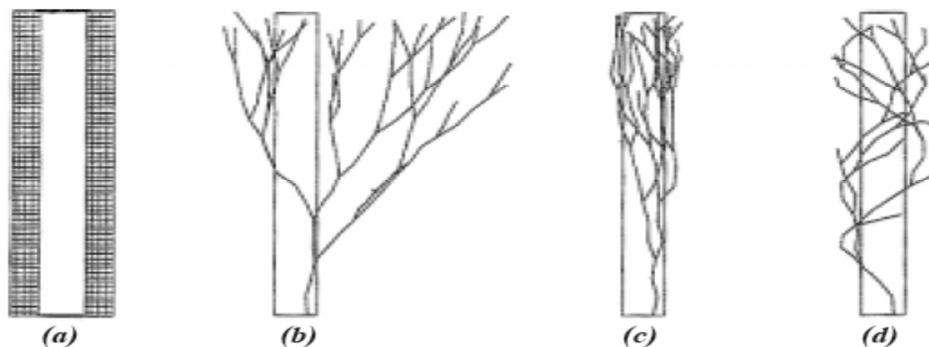
**Figure 35.** *Billboard en croix pour donner une impression de volume.*

### **2.2.2 Les modèles dans un espace de voxels**

Ned Green a publié une nouvelle méthode pour modéliser les plantes dans un espace discret [Gre89], l'utilisation d'un espace de voxels nous permet de simplifier un certain nombre d'opération :

- Les test d'intersection : Un simple test suffit à déterminer si une position dans l'espace est occupée ou non.
- En parcourant les voisins d'un voxel, on détermine rapidement les objets les plus proches.

Green utilise des arbres binaires engendrés par un processus stochastique dont l'un des moyens de contrôle est un retour de la géométrie sur la topologie avec pour conséquence une limitation de leur évolution. Cet arbre ne semble pas codé explicitement au cours de la modélisation. Le modèle géométrique conçu par Green est grandement influencé par les particularités de l'espace de travail utilisé: un espace Voxel est l'équivalent en trois dimensions d'une image bitmap en deux dimensions. Un objet (ligne, polygone, polyèdre) sera représenté comme un ensemble de voxels occupés. L'environnement proche d'un objet étant alors formé des couches successives de voxels voisins de ceux occupés par l'objet.



**Figure 36.** (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].

La représentation par espace de voxels permet de gérer efficacement la détection de collision entre les branches, les feuilles et avec l'environnement.

L'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 [Cc97].

### 2.2.3 Les modèles à base de système de particules

Cette méthode a été développée par Reeves [Ree83], elle a été utilisée à l'origine pour simuler des nuages, du feu et de l'eau son principe consiste à représenter un objet complexe par des primitives plus simple, plus petites mais très nombreuses.

Généralement ces primitives sont appelés particules, les attributs des particules sont les suivantes : la position dans l'espace, le vecteur de vitesse, la forme, la couleur l'opacité et l'âge de la particule.

Reeves et Blau [RB85] 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'ensevelissement est calculée à partir de sa position dans l'arbre, de la forme générale de l'arbre et de la position du soleil [Cc97].

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.



**Figure 37.** *Un exemple de paysage généré par des systèmes de particules.*

### **3 Bilan et Conclusion**

Nous avons présenté dans ce chapitre les différentes méthodes permettant de modéliser les forêts, nous avons ainsi fait le constat que modéliser un environnement naturel n'est pas une tâche aisée du fait de la complexité et de la quantité des éléments qui le composent. Dans le cadre de la modélisation d'une forêt de synthèse, on distingue deux grandes catégories de modèles :

Les approches structurelles qui tentent de construire une topologie la plus précise et la plus fidèle possible de la structure ramifiée. Ces méthodes peuvent être complexes et difficiles d'accès, mais elles génèrent des objets très proches des végétaux réels.

Les modèles géométriques sont des modèles intéressants car ils sont simples et permettent de générer facilement et automatiquement des niveaux de détails, basée entièrement sur la description de paramètres géométriques de la structure ramifiée.

Le principe de modèle à base grammaire est de représenter un arbre par une chaîne de caractères. Chacun de ces caractères est un élément de la plante. En partant d'un ensemble initial d'éléments et en lui appliquant plusieurs règles de production, une plante est obtenue après un certain nombre d'itérations.

Le modèle est facilement implémentable et extensible cependant il est difficile à contrôler et nécessite une certaine connaissance morphologique.

Les approches impressionnistes s'intéressent plus à l'aspect visuel du végétal plutôt qu'à la précision et le réalisme des structures produites.

On retrouvera la plupart du temps les méthodes à base de textures dans des applications dites « temps réel » qui nécessitent une grande rapidité de traitement que l'on obtient au détriment de la précision.

Le résultat de la modélisation et du rendu des arbres en utilisant les systèmes de particules 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.

L'objectif de notre mémoire étant la représentation de forêts en temps réel, nous étudier dans le chapitre suivant, les différentes techniques, nous allons en particulier nous intéresser aux techniques de représentation à base d'images.

## **Chapitre III :**

# **REPRESENTATION A BASE DE BILLBOARDS DES FORETS EN TEMPS REEL**

---

# REPRESENTATION A BASE DE BILLBOARDS DES FORETS EN TEMPS REEL

---

## 1 Introduction

Le rendu de scènes naturelles a fait l'objet de recherches variées étant donné ses nombreuses applications. Le but est de créer des images similaires à ce que l'on peut voir dans la réalité. Le principal obstacle est la complexité, les scènes naturelles sont très détaillées et difficiles à modéliser, elles demandent un temps de rendu important et beaucoup de mémoire, nous détaillerons dans ce chapitre les versions temps réel des forêts.

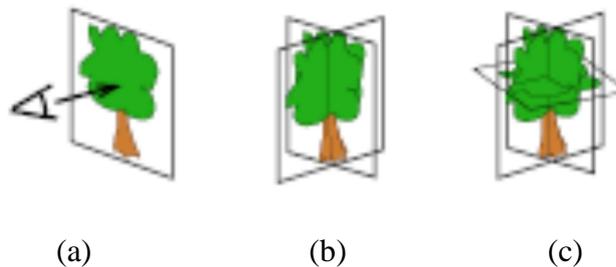
## 2 Les techniques d'imposteur temps réel pour les forêts

### 2.1 Les approches textures

L'utilisation d'images dans le rendu trouve son origine dans le texturage de polygones ce qui permet d'obtenir plus de détail sans augmenter la complexité de la géométrie, c'est la technique la plus connue, introduite par Catmull en 1974 [Cat74]

Les textures étant plaquées sur les polygones après leur projection dans le plan image, cette opération n'utilise que des calculs en 2 dimensions, ce qui fait que le coût en temps de calcul engendré est très faible par rapport aux opérations de projection, voire même négligeable par rapport au temps de calcul qui aurait été nécessaire pour rendre un objet en le modélisant directement par la géométrie.

Pendant longtemps cette technique a été utilisée pour représenter les arbres, mais avec l'arrivée des cartes graphiques 3D, les techniques de rendu par imposteurs sont imposées. Parmi les techniques d'imposteurs la technique dite *billboarding* qui est considérée comme le meilleur choix dans la génération des images dans de nombreuses applications comme les jeux vidéo, simulateurs ....



**Figure 38.** (a) *billboard classique*, (b) *2 billboard en croix*, (c) *3 billboard croix*.

La technique de *billboarding* consiste à représenter un arbre par un simple polygone rectangulaire sur lequel on plaque une texture, cette texture qui est une image semi-transparente de l'arbre tracé toujours face à l'observateur. Ces billboards perdent de leur efficacité si la caméra s'en approche trop et ils sont souvent victimes d'un effet « pancarte » qui trahit leur manque d'épaisseur. [OGL98]

Pour remédier en partie à l'impossibilité de se déplacer librement dans la scène, une autre représentation consiste à utiliser deux polygones semi-transparentes plaqués en croix avec un angle de  $90^\circ$ , le problème pour cette représentation est que si on observe l'arbre de dessus on risque de voir les arbres sur leurs tranches pour cela on peut utiliser une troisième texture représentant l'arbre en vue de dessus.

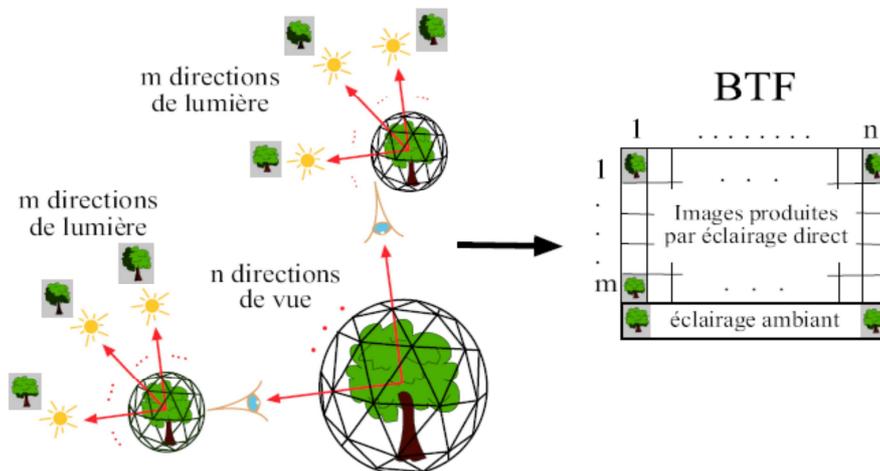
## 2.2 Modèle hiérarchique pour le rendu interactif de forêts

Meyer et al [MEY01] décrit une méthode pour rendre les forêts en temps réel cette méthode donne des images de qualité avec des effets complexes comme l'illumination, l'auto-ombrage et l'ombrage, en tenant compte du mouvement du soleil. Il consiste à construire une

fonction bidirectionnelle de textures associé aux billboards sur laquelle se calque une hiérarchie des cubes de visibilité (VCM) pour l'ombrage.

Le principe d'une fonction bidirectionnelle de texture (BTF) est d'associer une image de l'objet à chaque couple direction de vue plus direction de lumière, tout comme une BRDF associe une couleur à un couple de directions de vue et de lumière.

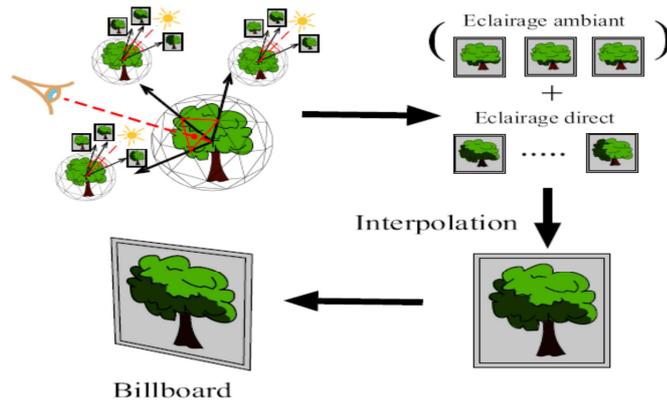
Pour le calcul, il considère  $n$  directions de vue, et pour chacune de ces directions de vue il considère  $m$  directions de lumière (figure 39). À chacun de ces couples (directions de vue, directions de lumière), nous associons une image de l'objet, composée de la couleur (RGB) et de l'opacité (alpha).



**Figure 39.** Construction d'une BTF.

L'introduction de cubes de visibilité pré-calculés permet de gérer l'ombrage sur les objets de la scène et autorise un déplacement interactif de la source de lumière.

Lors de la phase du rendu, à partir de la direction de vue et de la direction de lumière, il extrait les images de la BTF. Ces images sont interpolées pour former le billboard, et grâce à la BTF l'illumination et l'auto-ombrage sont encodés directement dans l'image : aucun autre calcul est nécessaire lors de la phase de rendu.

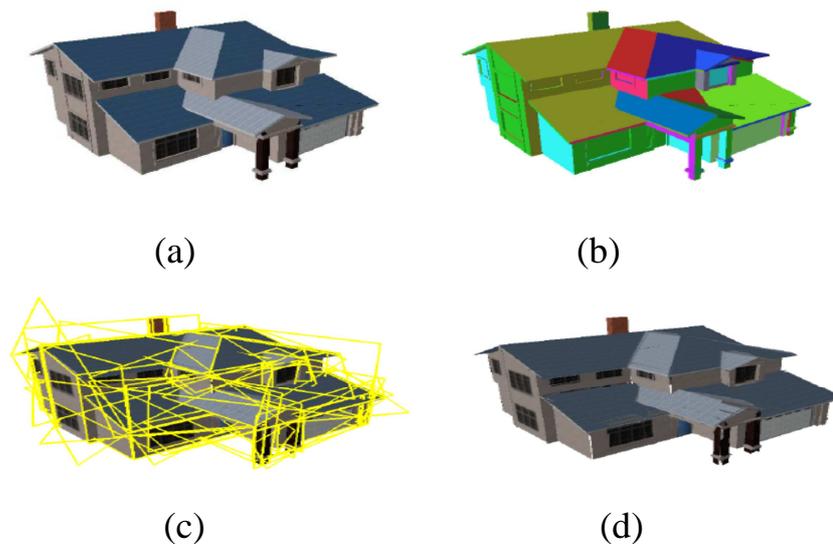


**Figure 40.** Interpoler de manière pondérée des images afin d'obtenir une représentation proche du billboard.

Cette méthode permet de représenter les scènes complexes en temps réel, cependant cette représentation présente quelques limites, un effet de flou apparaît à cause de l'interpolation et la faible résolution des images.

## 2.3 Les nuages de panneaux d'affichage (billboard clouds)

Un nuage de panneaux d'affichage est un ensemble de polygones (ou panneaux) texturés, partiellement transparents, avec des tailles, orientations et résolutions de texture indépendantes (figure 41). Typiquement, les billboards sont employés pour représenter des objets tels que des nuages ou des arbres, très difficiles à modéliser explicitement [DEC02].



**Figure 41.** Exemple d'un nuage de panneaux : (a) modèle original (1,960 polygones) (b) rendu en utilisant une couleur par panneau (c) Vue de 52 panneaux texturés (d) nuage de panneaux rendu.

Le principe de la méthode de Décoret et al [DEC02] est de remplacer un objet par la combinaison de plusieurs billboards représentant cet objet sous différents angles. Leur méthode permet de répartir automatiquement un certain nombre (fixé par l'utilisateur) de plans recouvrant l'objet de façon optimale. Cet ensemble de billboards est simplement affichés lors du rendu, les plans se superposent et se combinent à l'aide de la composante de transparence des textures.

Beaucoup de techniques ont employé la notion de polygones multiples, partiellement transparents pour accélérer l'affichage de modèles complexes. En effet, ces techniques peuvent être vues comme des cas spéciaux des nuages de panneaux [DEC02].

Les nuages de panneaux sont fortement efficaces dans la simplification de modèles complexes avec des textures multiples vers quelques dizaines de polygones texturés. La représentation de nuage de panneaux permet alors un rendu très rapide, ou un test de collision efficace. La qualité visuelle des modèles simplifiés est très haute et contrôlée par l'utilisateur. Le problème principal de cette méthode est que l'éclairage ne peut varier, du fait de l'emploi de placage de texture dépendant du point de vue.

## 2.4 Billboards volumiques

La technique des billboards volumiques est une extension de la représentation classique utilisée pour le rendu des scènes 3D complexes en temps réel, elle utilise des images volumiques d'un objet stocké dans des texture, 3D au lieu des images 2D comme pour billboard classique.

Les billboards volumiques offrent un effet de parallèle des objets dans toutes les directions sans artefacts et avec l'amélioration anti-aliasing des objets éloignés. Les objets peuvent avoir de la transparence et peuvent être arbitrairement distribués dans scène 3D.

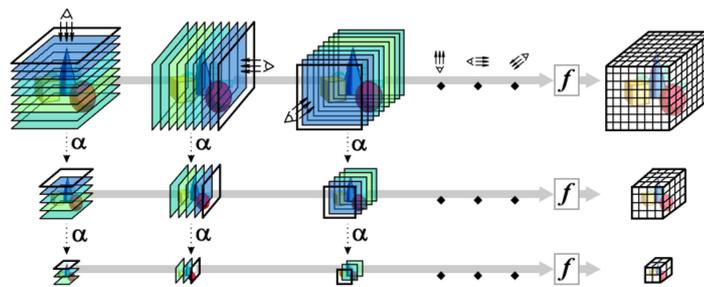
L'algorithme gère correctement la transparence entre les multiples objets superposés [DF09]. Cette technique consiste dans un premier temps à associer la texture 3D des données volumiques à leur volume englobant (prisme triangulaire), de sorte que les sommets du prisme correspondent aux sommets de la texture 3D, l'ensemble de prisme triangulaire sous forme de cellules positionné arbitrairement dans la scène 3D.

### **Génération de volume de données**

le volume de donnée représentant l'objet est créé en mode offline puis il est stocké dans une texture 3D MIP-mapped, le processus de construction veille à ce que l'information dépendant de vue est conservé aussi que possible.

La voxélisation d'un objet est réalisée par le rendu de 6 couche de tranches de l'objet sur les axes aligner, Chaque couche de tranches est obtenue en ajustant une caméra orthographique sur l'objet sur l'une des 6 directions (+ X,-X, Y +, - Y + Z,-Z).

A partir de ces 6 couches il produit 6 volumes 3D. Chaque volume est MIP-map de façon indépendante. Pour chaque niveau de MIP-map, les 6 volumes correspondants sont ensuite combinés en un seul volume, ces volumes forment finalement la 3D MIP-map texture de l'objet.



**Figure 42.** Un volume mip-map construit à partir de 6 directions de vue.

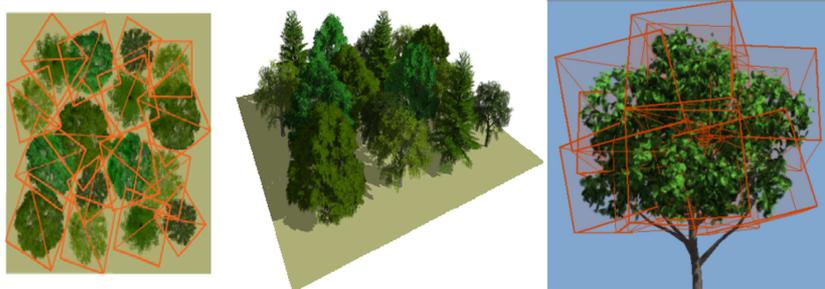
### **Le rendu de billboard**

A l'exécution, les panneaux d'affichage volumiques sont rendus avec un algorithme volumique à base de tranche qui rend tous les panneaux d'affichage volumique avec la transparence à la fois correctement et sans exiger de tri.

À cet effet, l'algorithme génère des tranches parallèles à l'écran, arrière et avant par rapport à la caméra. Pour chaque tranche, on calcule tous les polygones qui correspondent à l'intersection de la tranche donnée avec toutes les cellules.

Une valeur de texture (une couleur) peut être associée en tout point d'intérêt de la surface d'une tranche. On obtient donc des tranches de textures et la qualité du rendu volumique peut être ajustée en adaptant la distance entre les tranches avec le rééchantillonnage adéquat. Ensuite en rendant les plans en ordre back-to-front, l'opération de compositing ou de mélange

(blending) appropriée est réalisée, pour afficher la bonne couleur dans chaque pixel (encore géré par le GPU). La figure 43 présente quelques résultats obtenus par cette méthode.



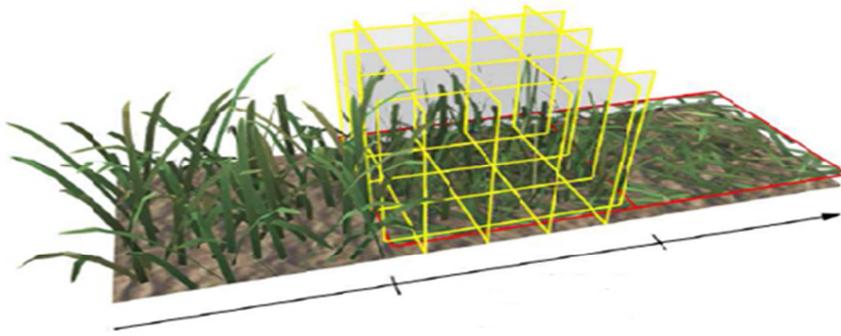
**Figure 43.** Arbres présentée par la méthode : *Billboard volumique* [DF09].

## 2.5 Rendu Réaliste de Scènes Naturelles

Kévin Boulanger [Kev08] décrit une méthode pour rendre les scènes naturelles en temps réel avec un éclairage global dynamique. Il consiste à étudier l'herbe et les arbres et propose une approche du rendu d'herbe temps réel qui offre un éclairage dynamique avec ombres et un effet de volume réaliste, le tout sans effet de crênelage, son système de niveaux de détail combine une méthode rendu basée géométrie avec une méthode basée volume. Cela lui permet de rendre de très larges surfaces herbeuses rapidement.

Pour gérer les niveaux de détails, il définit deux types de blocs de base : un ensemble de brins définis avec de la géométrie dans un rectangle ou un carré, et un ensemble de tranches alignées et texturées avec des images semi-transparentes (figure 44). Ces deux approches offrent un bon effet de parallaxe vu depuis n'importe quel angle, les blocs d'herbe ne paraissent pas plats.

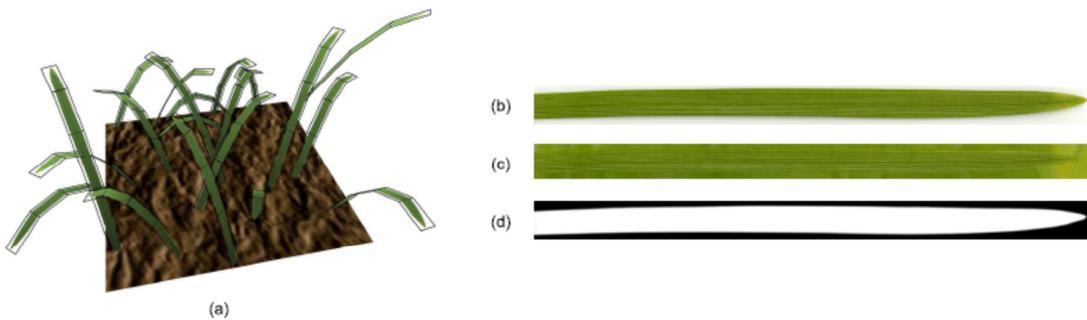
Il utilise la distance depuis l'observateur pour choisir le niveau de détail à utiliser. L'herbe définie par géométrie est utilisée au plus près de l'observateur, les tranches alignées à distance intermédiaire, la tranche horizontale étant la seule utilisée pour les surfaces les plus éloignées.



**Figure 44.** Les trois niveaux de détail, choisis en fonction de la distance de l'observateur.

L'herbe modélisée par géométrie est utilisée pour les surfaces proches de l'observateur et pour générer les données des tranches alignées du second niveau de détail.

Des chaînes de quadrilatères texturés avec un canal de transparence sont employées pour permettre de définir précisément la forme du brin (figure 45).

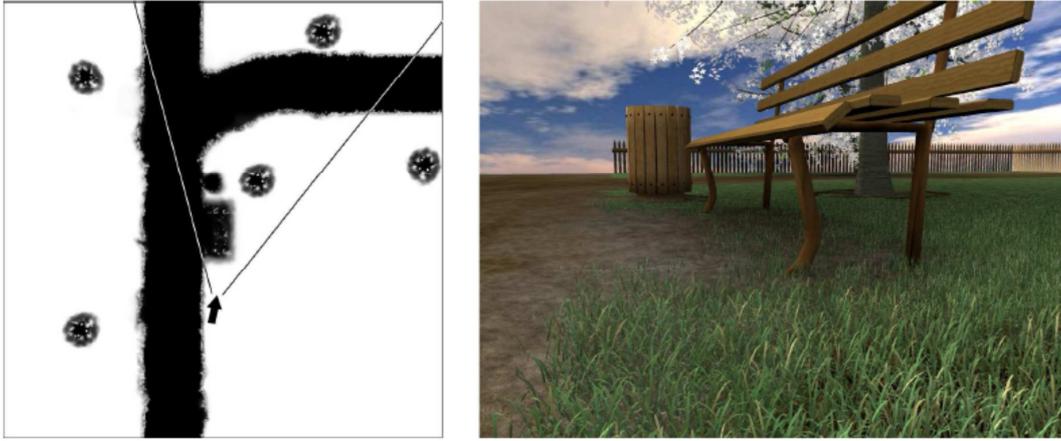


**Figure 45.** (a) Brins d'herbe avec chaînes de quadrilatères texturées. (b) Scan d'un brin réel. (c) Canal de couleur de la texture, la bordure est dupliquée pour éviter les bordures blanches lors du rendu. (d) Canal alpha de la texture pour définir la forme du brin.

L'utilisation de rendu volumique pour l'herbe à distance intermédiaire de la caméra, là où le rendu géométrique est trop coûteux. L'avantage de cette approche est le bon effet de parallaxe : quand l'observateur se déplace, l'herbe paraît toujours être en 3D.

Le terrain est subdivisé en une grille régulière et une instance du volume élémentaire est placé dans chaque cellule de la grille. Le volume élémentaire représente un bloc de quelques milliers de brins.

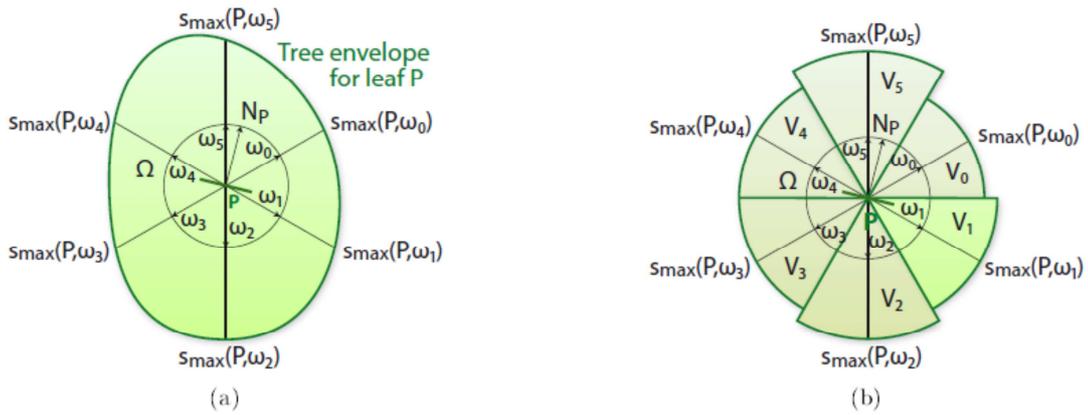
Il est souvent difficile de gérer la densité précisément dans les applications temps réel, il faut gérer la densité brin par brin, pour chaque point du sol. Au lieu de caractériser chaque brin individuellement, une carte de densité (figure 46) est utilisée pour donner la densité locale pour chaque point du sol.



**Figure 46.** Densité variable de brins définie par une carte de densité. L'image de gauche est une carte qui permet d'obtenir le résultat de l'image de droite.

Il modélise le feuillage d'un arbre avec une enveloppe de volume  $V$ . Cette enveloppe contient un ensemble de feuilles réparties uniformément dans l'espace, avec une densité  $\rho$ , une aire par feuille  $A$  constante et une distribution uniforme des normales  $N_P$ . Au lieu de définir l'enveloppe une seule fois pour tout l'arbre, il utilise une approximation de l'enveloppe du point de vue de chaque feuille :  $N_{dir}$  distances  $s_{max}(P, \omega_j)$  sont définies entre le centre de chaque feuille  $P$  et l'enveloppe pour  $N_{dir}$  directions d'échantillonnage  $\omega_j$  distribuées uniformément sur la sphère unitaire  $\Omega$  (figure 47 (a)). L'évaluation de ces distances est réalisée lors d'un précalcul rapide. L'ensemble des directions d'échantillonnage  $\omega_j$  est le même pour toutes les feuilles. Cela permet l'évaluation de la luminance incidente par arbre et non par feuille. Un sous-volume  $V_j$  du volume total  $V$  est associé à chaque direction  $\omega_j$ , avec un angle solide  $\Omega_j$ . Les sous volumes  $V_j$  ont une forme pyramidale dont le sommet est en  $P$  et dirigés par  $\omega_j$  (figure 47 (b)). Nous avons  $V = \sum_{j=1}^{N_{dir}} V_j$ , le nombre de feuilles dans chaque sous-volume  $V_j$  peut-être déterminé analytiquement :

$$N_{leaves_j} = \rho V_j = \rho \frac{4\pi}{N_{dir}} \frac{(s_{max}(P, \omega_j))^3}{3}$$



**Figure 47.** (a) Paramétrisation de l'enveloppe d'un arbre depuis la feuille centrée en P.  
(b) Ensemble de sous-volumes  $V_j$  équivalent.

La figure 48 présente quelques résultats obtenus par cette méthode.



**Figure 48.** Forêts présentée par la méthode de KEVIN [Kev08].

### 3 Bilan

Une technique largement utilisée dans les jeux vidéo est la technique de billboardage, elle consiste à remplacer des objets de géométrie complexe par un plusieurs imposteurs plats et semi-transparents texturés avec une image de l'objet.

L'avantage de cette technique est :

- d'offrir un modèle simple à mettre en œuvre et très efficace, il est alors possible d'avoir de nombreux arbres pour un faible coût.

Par contre :

- le coût mémoire augmente vite si l'on veut avoir une diversité d'apparence des arbres.
- l'impossibilité de changer l'illumination ou l'ombrage d'un tel modèle parce que celle-ci est fixée dans l'image.

On retrouvera la plupart du temps ces représentations dans des applications dites « temps réel » qui nécessitent une grande rapidité de traitement que l'on obtient au détriment de la précision.

L'avantage principal de nuages de panneaux [DEC02] consiste en ce qu'aucune information topologique (comme la connectivité de polygone) n'est exigée. De plus la complexité visuelle de nuage de panneaux résulte principalement de la texture le principal inconvénient de cette méthode est :

- la lenteur de la phase de construction, qui utilise une transformée de l'espace objet similaire à la transformée de Hough pour calculer les plans approximant le mieux l'objet.
- De plus les temps de rendu et la consommation en mémoire sont relativement élevés.
- certains artefacts peuvent apparaître principalement dans les régions courbes à cause de l'approximation de la géométrie.

L'inconvénient principal de la représentation volumique de billboard réside dans la taille des données à stocker.

**Chapitre IV :**

**Le Modèle Proposé**

---

# **Le Modèle Proposé : Génération des Forêts en Temps Réel**

---

## **1 Introduction et Motivations**

Lorsque l'on s'intéresse à la génération de végétaux virtuels, la génération par L-système s'impose souvent comme technique de référence. Présentés en 1968 par Aristid Lindenmayer.

Les l-systèmes se basent sur les notions de grammaire. A partir d'un mot de départ et d'un ensemble de règles, un nouveau mot est généré par l'application des règles sur le mot de départ. Les règles s'appliquent généralement de façon récursive, ce qui va souvent donner un aspect auto-similaire aux mots générés. Or si l'on observe des plantes, on s'aperçoit qu'elles disposent de fortes propriétés d'autosimilarité et d'une grande diversité, que ce soit au niveau de la géométrie, on aura par exemple des formes cylindriques pour les branches des arbres, et des formes plutôt planes pour les feuilles, d'où l'idée d'utiliser les L-systèmes.

D'autre part les paramètres géométriques peuvent être intégrés dans les règles de production pour simuler par exemple les angles d'embranchement ou les rapports des longueurs des branches.

Nous voulons concevoir un nouveau modèle permettant le rendu de forêt en temps réel tout en étant le plus réaliste possible. Pour réaliser cela nous nous sommes attachés à combiner les avantages des différents modes de représentation de la scène en développant une technique de rendu hybrides. Nous allons diviser la représentation d'arbre en deux entités distinctes :

- les feuilles qui seront représentées par des billboards semi-transparents.
- le tronc et les branches qui seront représentés par une structure L-système.

Les principales motivations de choisir un modèle hybride pour ce type des scènes est :

- une grande diversité au niveau de la géométrie : le branche et l'écorce ont une forme cylindrique, les feuilles ont une forme plane , les fruits ont une forme plus complexe .
- une grande diversité des échelles : les modèles les plus détaillés sont afficher au premier plan, et des modèles de plus en plus dégradées en fonction de l'éloignement .

Cette diversité au sein de la géométrie et des échelles utilisées incite fortement à penser qu'il serait très profitable de pouvoir utiliser une technique hybride.

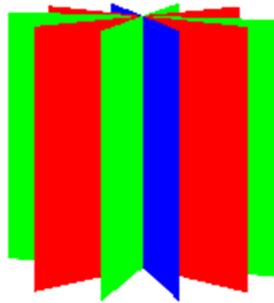
## **2 Description du modèle proposé**

Pour traiter la complexité posée par les forêts, nous avons choisi un modèle hybride, notre représentation est constituée de deux types de modèle, les arbres qui se trouvent loin de l'observateur sont représentées par la technique de billboard, par contre les arbres du premier plan sont générées par un modèle hybride L-système-billboard.

### **2.1 Génération des arbres avec la technique de billboard**

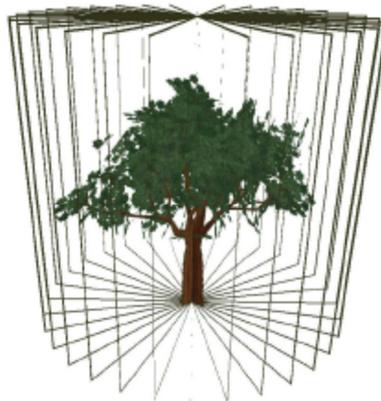
La technique de billboard consiste à modéliser chaque arbre par plusieurs quadrilatères texturés semi transparents placés en forme de croix.

Notre modèle de billboard est une combinaison de deux techniques de billboard : les billboards orientés vers le centre de la caméra, ils représentent mieux les objets présentant une symétrie cylindrique, les billboards croisés constituées de plusieurs plans en croix (indépendants de la caméra), ils permettent d'offrir un meilleur effet de perspective et de volume.



**Figure 49.** *La structure de billboard utiliser.*

L'idée de base de technique billboard orientés vers le centre de la caméra, est de représenté un arbre par un quadrilatère sur lequel en plaque une image stocké dans une texture semi-transparente, à chaque calcul d'image la position de billboard est mise à jour pour être face à l'observateur pour cela il peut tourner autour d'un axe ou d'un point de symétrie. La technique de billboard croisé consiste à représenter l'objet avec un ensemble de quadrilatères positionnés en croix pour donnée un effet de volume, le gros avantage de cette technique est qu'elle ne requiert pas beaucoup plus de calcul pour l'affichage.



**Figure 50.** *Plusieurs billboard croisés.*

Le placage de texture consiste à affecter une image à un polygone, de manière générale, plaquer une texture sur une primitive consiste à associer à chaque point de sa surface, des coordonnées  $(u,v)$  dans une image bidimensionnelle.  $(u,v)$  sont appelées les coordonnées de texture. La couleur de la primitive en un point est alors la couleur de l'image en  $(u,v)$ .

### 2.1.1 Placage de texture et transparence

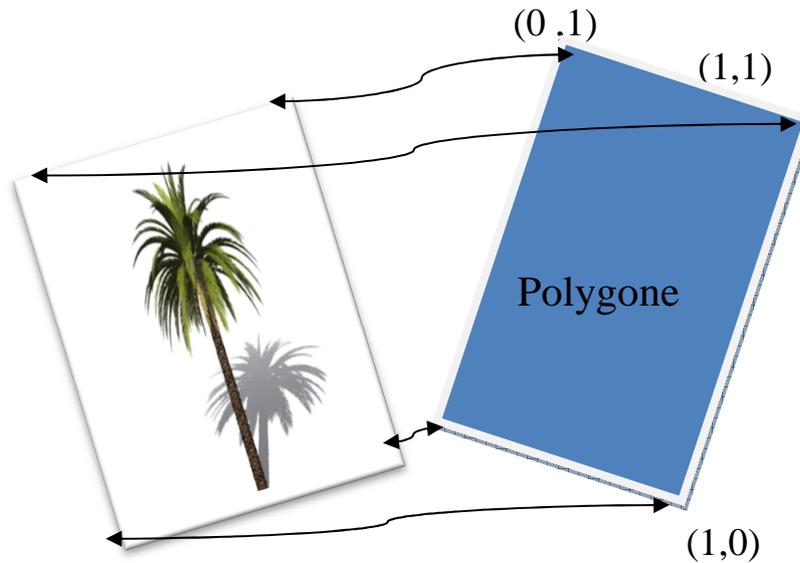
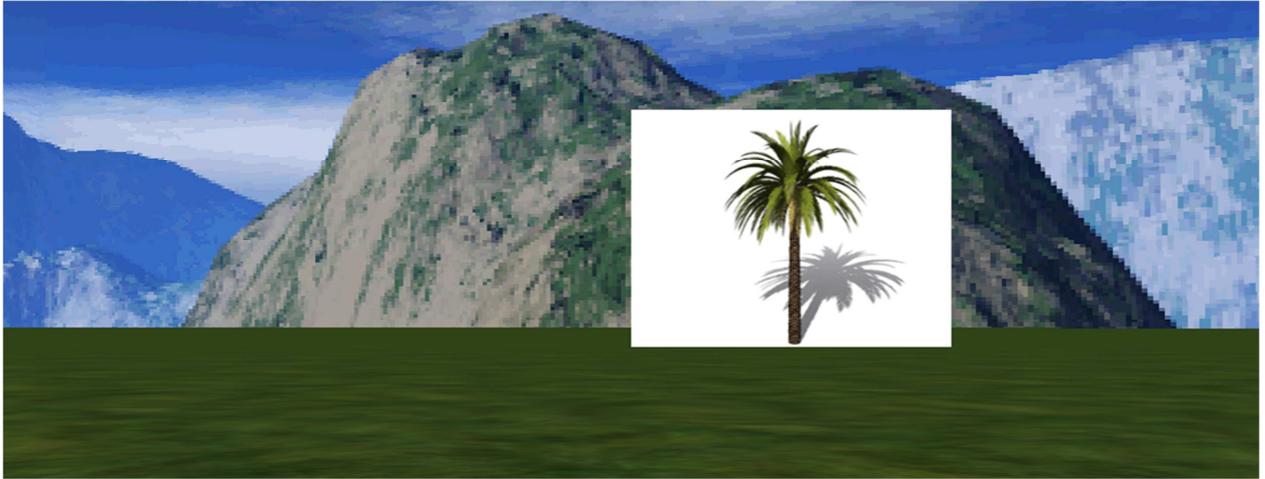


Figure 51. Le placage de texture.

- Il faut activer le texture mapping par la commande `glEnable(GL_TEXTURE_2D)` et le désactiver (avec `glDisable`) après l'affichage des polygones
- De plus, chaque `glVertex3f` doit être précédé d'un appel à `glTexCoord2f(u,v)` où  $u$  et  $v$  sont des coordonnées textures comprises entre 0 et 1. Ces coordonnées correspondent à des positions dans l'image.

Il nous reste cependant le problème des trous de l'image qui nuit au réalisme de la scène. Pour résoudre ce problème, on utilisera la composition de couleurs (l'alpha-blending).



**Figure 52.** *Le placage de texture d'arbre sur le billboard.*

La méthode d'Alpha Blending permet d'afficher des pixels semi-transparents, c.à.d. de mélanger les fragments composants l'image d'un objet au fond de la scène ou à un autre objet. de façon à représenter une transparence partielle, nous allons utiliser une autre image texture alpha pour afficher un plan semi-transparent au travers duquel on voit la scène.

La texture alpha est une couche supplémentaire que l'on peut rajouter sur une image afin que certaine partie soient transparentes, et d'autres non.

- La partie noir sera entièrement transparente
- La partie blanche sera entièrement opaque

Il faut activer la transparence dans OpenGL. Ceci se fait grâce à la commande `glEnable(GL_BLEND)`, il existe différentes fonctions de mélange. Celles ci sont spécifiées par un appel à `glBlendFunc()`. l'opération de compositing ou mélange de couleur consiste à initialisé le compositing sous forme RGBA

$$\mathbf{C}_{\text{dest}} = (0.0, 0.0, 0.0, 0.0)$$

Le compositing permet de tracer les objets transparents du plus profond au moins profond, est donné par la formule suivante :

$$\mathbf{C}_{\text{dest}} = (1 - \alpha_{\text{src}}) \cdot \mathbf{C}_{\text{dest}} + \alpha_{\text{src}} \cdot \mathbf{C}_{\text{src}}$$

$$\mathbf{C}_{\text{dest}} = \mathbf{C}_{\text{dest}} + (1 - \alpha_{\text{dest}}) \cdot \alpha_{\text{src}} \cdot \mathbf{C}_{\text{src}}$$

$$\alpha_{\text{dest}} = \alpha_{\text{dest}} + (1 - \alpha_{\text{dest}}) \cdot \alpha_{\text{src}}$$

Où  $\mathbf{C}_{\text{dest}}$  : désigne la couleur finale calculée, celle qui se trouvera dans un color buffer (tampon mémoire qui contient des couleurs pour afficher l'image correspondante selon la résolution).

$C_{src}$  : désigne la couleur actuelle.

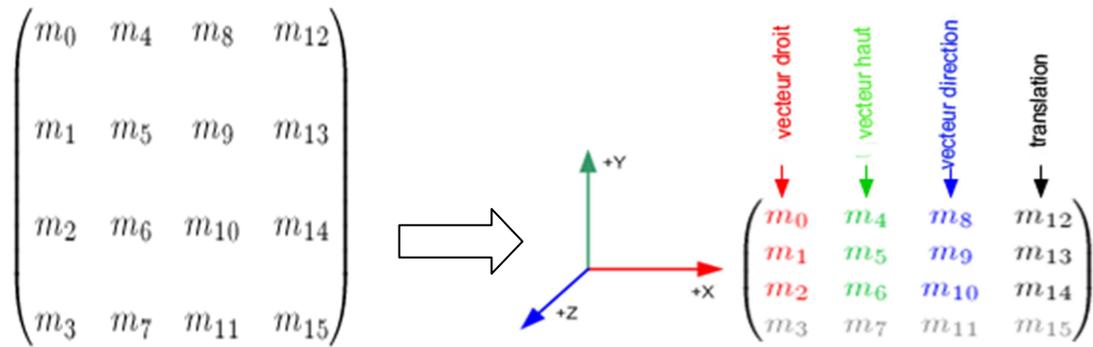
Nous allons utiliser l'opération alpha-test, qui élimine la partie complètement transparente entre l'arbre et la bordure de la texture,



**Figure 53.** *Billboards avec transparence.*

### **2.1.2 Calcul de la transformation**

L'implémentation de la technique de billboardage avec la bibliothèque opengl se fait grâce à la matrice ModelView qui contient toutes les transformations géométriques : rotation, échelle et translation. Elle contient aussi les transformations nécessaires pour changer les coordonnées entre le repère local et le repère caméra. La matrice ModelView est un ensemble de valeurs sous forme d'un tableau ayant un certain nombre de lignes et de colonnes. Les trois éléments de la colonne droite ( $m_{12}$ ,  $m_{13}$ ,  $m_{14}$ ) sont réservés à la transformation de translation, l'élément  $m_{15}$  est l'élément de coordonnées homogènes qui est utilisé pour la transformation projective. Les trois séries d'éléments de la sous matrice 3\*3 ( $m_0, m_1, m_2$ ), ( $m_4, m_5, m_6$ ) et ( $m_8, m_9, m_{10}$ ) sont pour les transformations euclidienne et affine.



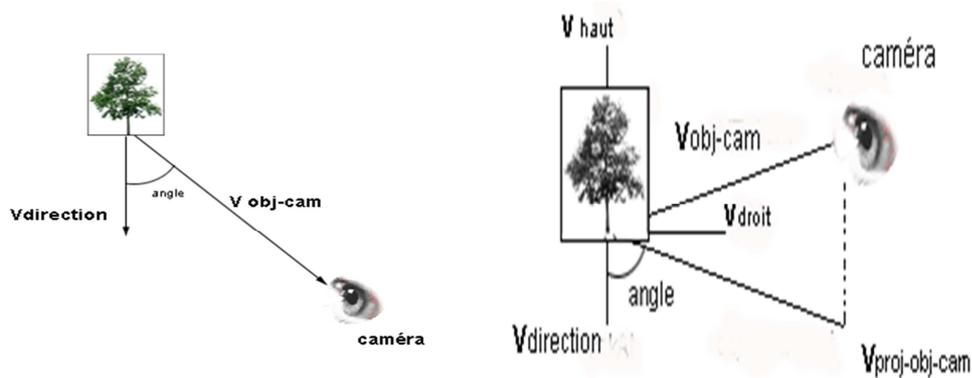
**Figure 54.** Extraction d'information.

A partir de la matrice ModelView, nous avons les vecteurs de direction de l'objet, le vecteur droit  $V_{\text{droit}} (m_0, m_1, m_2)$ , le vecteur haut  $V_{\text{haut}} (m_4, m_5, m_6)$  et le vecteur direction  $V_{\text{direction}} (m_8, m_9, m_{10})$ . Pour annuler toute rotation de billboard nous allons remplacer la sous matrice 3\*3 de la matrice ModelView par l'identité. Cette version de billboard est dite sphérique. Il existe une autre version dite cylindrique. Dans la version cylindrique, lorsque la caméra regarde vers le haut ou vers le bas le billboard ne bougera pas, le Billboard ne sera mis en rotation que si la caméra regarde à droite ou à gauche, ce type de billboard peut être appliqué aux arbres parce que les arbres ne se plient pas en arrière ou en avant quand la caméra regarde vers le haut ou vers le bas parce que le vecteur haut est fixe.

$$\begin{pmatrix} 1 & m_4 & 0 & m_{12} \\ 0 & m_5 & 0 & m_{13} \\ 0 & m_6 & 1 & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix}$$

**Figure 55.** Remplacement de la sous matrice par l'identité.

Ce que nous avons ici est un billboard face au plan perpendiculaire à la caméra, pour que le billboard pointe directement au centre de la caméra. Nous devons calculer la rotation de billboard manuellement. Pour des raisons de simplicité de calculs, nous devons mettre  $V_{\text{droit}} (1, 0, 0)$ ,  $V_{\text{haut}} (0, 1, 0)$  et  $V_{\text{direction}} (0, 0, 1)$ , en pratique cela signifie que l'objet est à l'origine de l'axe Z positif.



**Figure 56.** Représentation de billboard par rapport au caméra.

Pour faire une rotation de l'objet autour de l'axe  $V_{haut}$  avec une angle entre le vecteur direction et le Vecteur obj-cam, le vecteur objet caméra est obtenu en soustrayant la position de l'objet à partir de la position de la caméra. La projection de Vecteur  $V_{obj-cam}$  est obtenue par la mise de la composante de  $y$  à zéro, l'angle (définie en fonction de cosinus) est déterminée avec le produit scalaire entre le vecteur  $V_{direction}$  et le  $V_{proj-obj-cam}$ . Cependant, le cosinus ne suffit pas puisque  $\cos(\alpha) = \cos(-\alpha)$ , pour cela le calcul de produit vectoriel permet de déterminer l'angle unique. Dans le cas où l'angle est positive le vecteur produit vectoriel sera orienté dans la direction de l'axe  $Y$  positif sinon il sera orienté vers le l'axe  $Y$  négatif.



**Figure 57.** Rendu une forêt de 1140 arbres, chaque arbre représentée par un billboard plat.



**Figure 58.** Une scène de 1600 arbres, chaque arbre représentée par 8 billboards en croix.

## 2.2 Génération des arbres avec les L-système

### 2.2.1 Représentation d'arbre avec un L-système

La génération des plantes par L-système s'impose souvent comme une technique de référence. Présentés en 1968 par Aristid Lindenmayer [Lin68], les L-systèmes se basent sur les notions de grammaire. A partir d'un mot de départ et d'un ensemble de règles, un nouveau mot est généré par l'application des règles sur le mot de départ. Les règles s'appliquent généralement de façon récursive.

Les L-systèmes sont définis par une grammaire qui est une quadruplet  $G = \langle V, S, w, P \rangle$  où :

- **V** : l'ensemble des variables du L-Système.
- **S** : un ensemble de valeur constantes.
- **w** : est un mot paramétré non vide dit axiome.
- **P** : un ensemble de règles de reproduction des symboles de V.

Un mot généré par un L-système est constitué de plusieurs symboles, pour aboutir à représenter une arbre 3D à partir d'un tel mot nous devons faire une interprétation géométrique des symboles composant ce mot, notre représentation est inspirée d'une méthode d'interprétation beaucoup utilisée dans la modélisation réaliste des plantes cette méthode dite interprétation tortue.

Un état de la tortue est défini par un triplet  $(x, y, \alpha)$ , où les coordonnées cartésiennes  $(x, y)$  représentent la position de la tortue, et l'angle  $\alpha$  est la direction dans laquelle regarde la tortue. Étant donné un pas  $d$  et un incrément angulaire  $\alpha$ , la tortue répond aux commandes représentées par les symboles suivants :

F : Avancer d'un pas de longueur  $d$ . Une ligne est tracée entre l'ancien et le nouvel état de la tortue.

f : Avancer d'un pas  $d$  sans tracer de ligne.

+ : Tourner à gauche d'angle  $\alpha$

- : Tourner à droite d'un angle  $\alpha$

[,] : Les crochets permettent d'empiler et de dépiler l'état de la tortue à un instant donné, ce qui permet de partir sur la création d'une branche puis de revenir à sa racine pour continuer le tracé.

### 2.2.2 *Génération de la structure de l'arbre*

Notre méthode consiste en premier lieu à générer un squelette de l'arbre à partir du mot produit par un L-système puis la modélisation géométrique est effectuée au cours d'une seconde passe appelée "interprétation".

A partir d'un mot de départ et d'un ensemble de règles, un nouveau mot est généré par l'application des règles sur le mot de départ. Les règles s'appliquent généralement de façon récursive.

Pour modéliser un arbre, une signification est donnée à chaque caractère de l'alphabet, généralement une branche pour les caractères variables et une transformation pour les symboles constants.

Soit le L-Système  $FF[-F][+F]F+F$ , qui représente le squelette de l'arbre

Lettres: F

Constantes: [, ], +, -

Axiome: F

Règles:  $F \rightarrow FF[-F][+F]F+F$

Avec

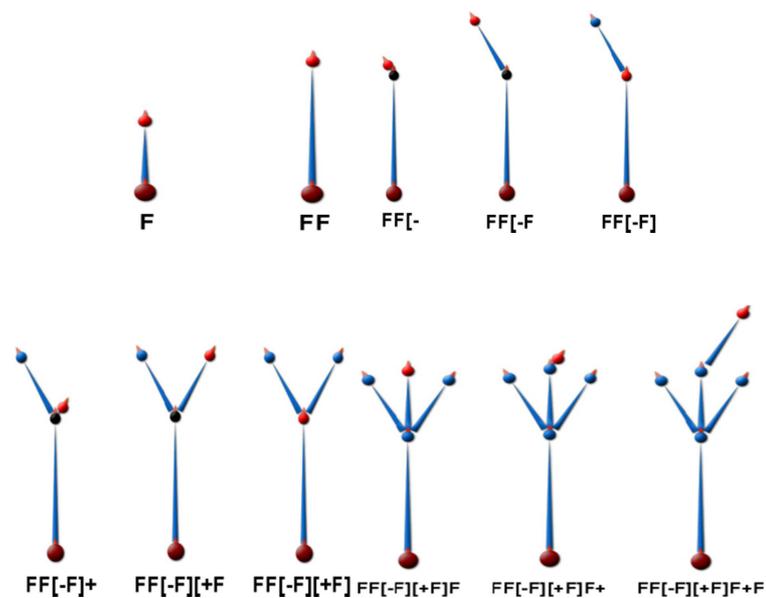
F : signifiant afficher une branche,

+ : faire une rotation  $R(30^\circ)$

- : faire une rotation  $R(-30^\circ)$

[, ] : sauvegarder, restaurer la position et l'orientation.

Le processus de génération d'un squelette à partir d'un mot généré par une grammaire L-système est comme suit :

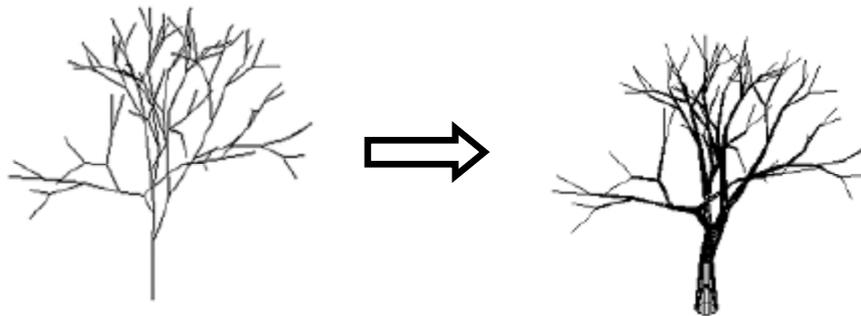


**Figure 59.** *Processus de construction de squelette à partir d'un L-système.*

Après la création du mot décrivant la topologie de l'arbre. Nous allons ensuite représenter l'arbre tridimensionnel à partir d'un tel mot. Pour cela il faut pouvoir donner une interprétation géométrique aux symboles qui le composent.

### 2.2.3 Interprétation géométrique

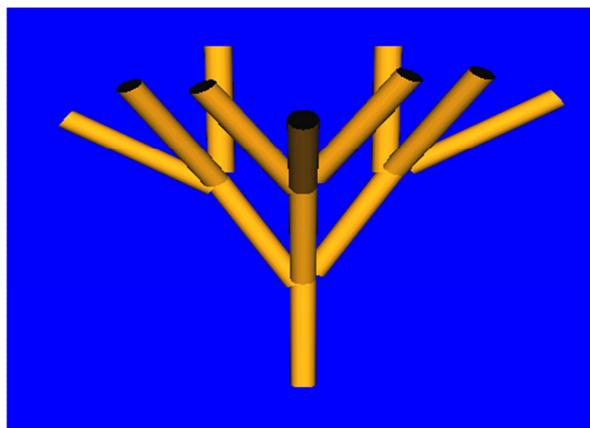
Partant du constat simple que les branches sont de forme plus ou moins cylindrique, la primitive de base que nous avons employée pour donner du volume à chaque segment de branche est le cylindre. Ainsi, en donnant un rayon à chaque segment du squelette de l'arbre, on a pu obtenir des résultats visuellement plus représentatifs qu'une simple représentation filaire.



**Figure 60.** *Interprétation géométrique d'une squelette.*

Pour assurer une connectivité de l'objet au final, quelques informations sont nécessaires :

- Il s'agit de la position des extrémités du segment pour éviter une discontinuité à la jonction de deux cylindres de rayons différents.
- Une information très importante est celle donnant la dimension des extrémités du cylindres, nous allons stocker le diamètre à chaque extrémité est le rayon de la base du cylindre.

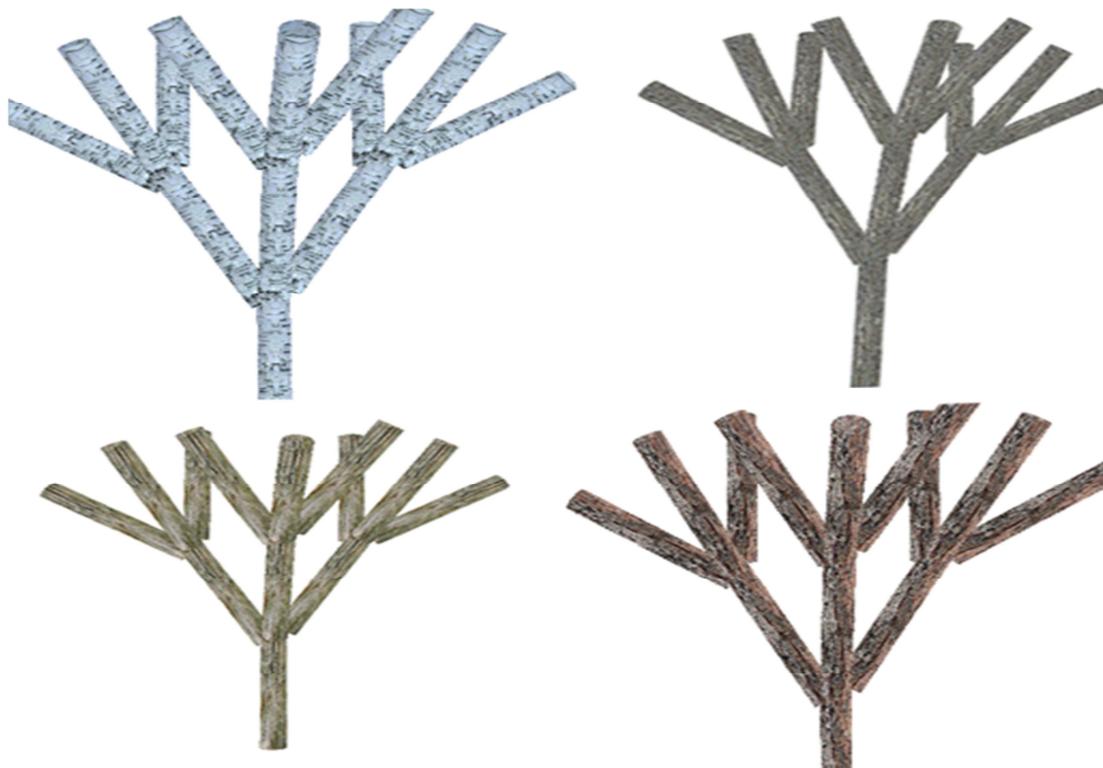


**Figure 61.** *Interprétation géométrique d'un squelette avec la primitive cylindre.*

Pour définir précisément la forme de l'écorce et les branches de l'arbre nous devons appliquer une texture 2D sur une forme cylindrique.

Plaquier une texture sur un cylindre consiste à trouver pour chaque point P de l'objet à texturer son correspondant dans l'image, une image caractérisée par un ensemble de pixels(R,G,B) un repère normalisé lui est associé.

La figure suivante montre quatre exemples de texture d'écorce que nous avons synthétisés.



**Figure 62.** Ecorce et branches d'un arbre texturé avec un placage de textures 2D.

Jusqu'ici nous n'avons traité que les branches et les écorces qui les recouvre. Cependant il reste encore un élément essentiel de l'arbre, qui apporte un poids substantiel au réalisme visuel : les feuilles.

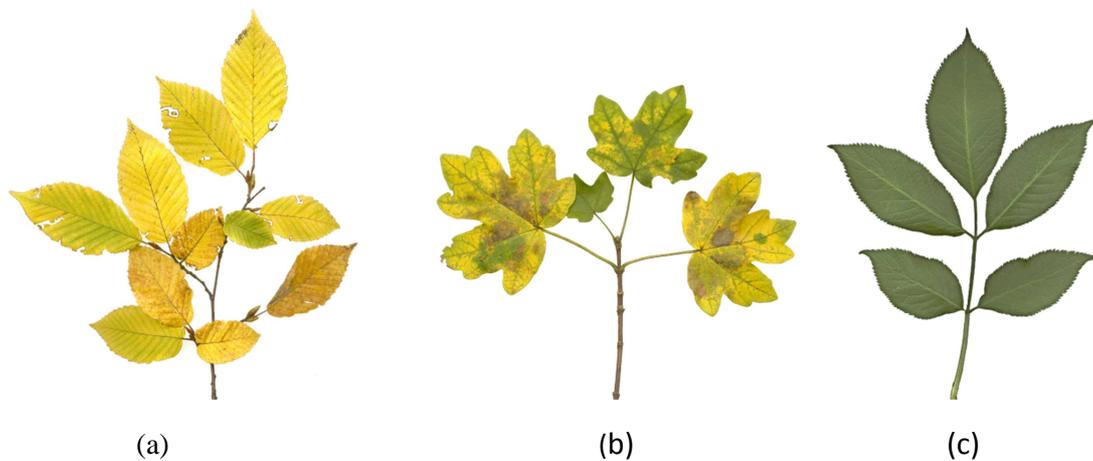
#### **2.2.4 Génération des feuilles de l'arbre**

Notre idée est d'insérer un caractère dans l'alphabet de L-système, l'interprétation géométrique de ce caractère est un billboard, un quadrilatère semi-transparent représente les petites branches et les feuilles de l'arbre.

Une opération de blending est nécessaire pour afficher ces quadrilatères semi transparents.

Cependant, les polygones doivent être triés de l'arrière vers l'avant, ce qui est une opération beaucoup trop coûteuse. Nous combinons donc l'alpha test avec le blending :

L'opération d'alpha test élimine la partie complètement transparente entre les feuilles et la bordure de la texture, le blending se charge de composer les pixels des bords des feuilles avec l'arrière-plan.

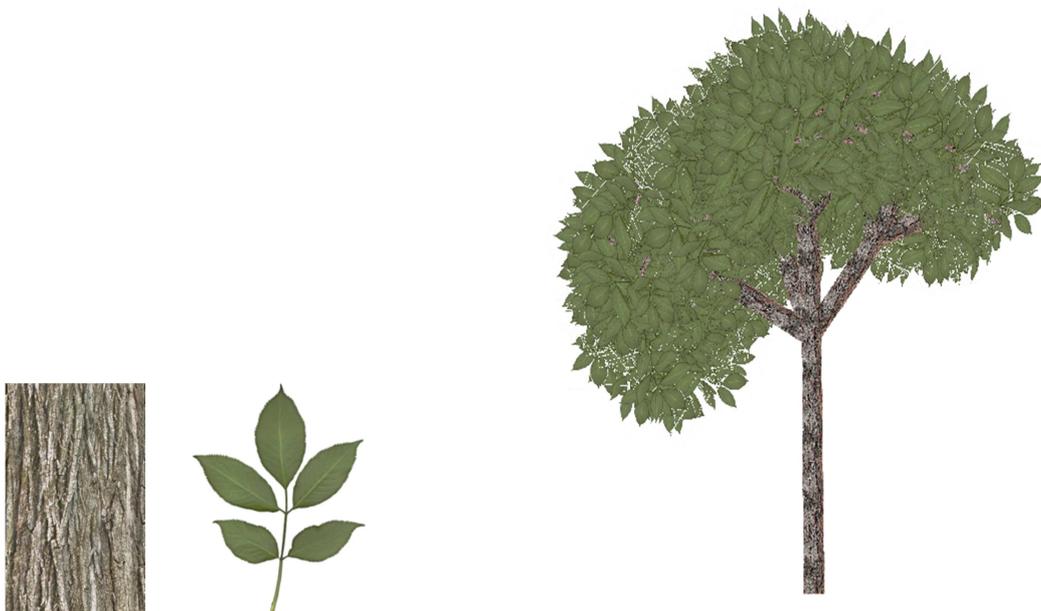


**Figure 63.** *Quelques exemples de texture 2D, illustrant la diversité des feuilles.*

Nous présentons ci-dessous deux exemple d'arbres générés avec divers types de textures de l'écorce et des feuilles différentes.



(a) *Arbre avec texture de l'écorce et type de feuille : Acer campestre (Origine Europe Centrale)*



(a) *Arbre avec texture de l'écorce et type de feuille : Populus nigra Italica (origine Lombardie Italie).*

**Figure 64.** Deux exemples d'arbres (a ) et (b) générés par notre système.



**Figure 65.** *Deux arbres avec des textures d'écorce et feuilles différent.*

### **3 Conclusion**

Le présent chapitre nous a permis d'exposer les fondements de notre méthode, dont l'objectif est de générer des forêts, composées d'arbre en temps réel.

La technique que nous avons proposé repose essentiellement sur l'hybridation de la géométrie et la représentation à base d'image, permettant de trouver un bon compromis entre l'aspect visuel et le temps d'excusion.

**Chapitre IV :**  
**Mise en œuvre, Résultats**  
**et Bilan**

---

# Mise en œuvre, Résultats et Bilan

---

## 1 Introduction

Ce chapitre a pour but de valider expérimentalement, à travers un ensemble de résultats, le modèle proposé dans le chapitre précédent.

Nous commençons par expliquer les bases de notre algorithme puis nous donnons plus d'informations sur chaque technique de représentation d'arbre utilisées. Enfin, nous présentons des résultats du rendu et les performances de l'algorithme proposé.

Les résultats sont obtenues en utilisant un Pentium Core 2 Duo 2.40 GHz équipé d'une mémoire 2.0 Go DDR et une carte graphique NVidia GeForce 9400 GT avec une mémoire de 512 Mo.

## 2 Les algorithmes utilisés

Notre approche de rendu consiste à rendre une forêt en temps réel, nous avons implémenté une technique de billboard pour les arbres de l'arrière plan.

Il faut faire en sorte que le billboard soit toujours tourné face à la caméra. Comme on sait qu'en réalité c'est la scène qui tourne pour se mettre devant la caméra, ceci grâce à la matrice ModelView, il faut faire en sorte d'annuler toute rotation du billboard : on remplace la partie

3x3 de la matrice ModelView par l'identité et on ne laisse que l'information de translation. Dans ce cas, on ne peut absolument pas faire une liste d'affichage.

$$\begin{bmatrix} 1 & 0 & 0 & a_{12} \\ 0 & 1 & 0 & a_{13} \\ 0 & 0 & 1 & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{bmatrix}$$

**Figure 66.** Remplacement de la sous matrice par l'identité.

La fonction qui permet d'annuler toute rotation du billboard est présentée ci-après

---

### Algorithme : d'annulation de rotation de billboard

---

Début

- Récupérer la matrice model view courante ;
- Modifier la matrice ModelView afin qu'il n'y ait plus aucune rotation ;

```

Pour ( pour i := 0,3 , pas 2 ) Faire
  Pour (pour j := 0,3 , pas 2 ) Faire
    Si ( i = j ) alors
      ModelView [i *4+j] ← 1 .0 ;
    Si non
      ModelView [i *4+j] ← 0 .0 ;
    Fin Si ;
  Fin Pour ;
Fin Pour ;

```

- Remettre la matrice ModelView après l'avoir modifiée ;
- fin ;
-

Après l'exécution de cette partie de l'algorithme qui annule toute rotation ou translation de billboard nous devons restituer la matrice Model view sauvegardée avec un dépilement de la pile de rendu.

La question qui se pose maintenant est Comment savoir vers où diriger la face du billboard ?

Et comment le billboard est orienté face au point de vue ? , nous avons besoin de connaître quelques propriétés comme :

- La position de l'objet.
- La position de la caméra.
- Le point vers lequel regarde la caméra

---

#### Algorithme de rotation de billboard

---

Début

- Projeter le vecteur objet\_camera sur le plan XZ ;
- Normaliser le vecteur objet\_camera ;
- Angle\_rotation ← produit scalaire (vecteur de vue , objet\_camera) ;

Si ( (angle\_rotation < 0,9999) et (angle de rotation > -0,9999)) alors

Effectuer une rotation de billboard suivant l'axe y est une angle de  
angle ← angle\_rotation

Fin si

Fin ;

---

Après la modélisation des billboards nous devons plaquer une texture lue à partir d'un fichier qui représente une image d'arbre sur ces billboards.

---

Algorithme chargement de l'image dans une texture

---

Début

- Ouvrir le fichier
- Vérifier s'il s'agit d'une image BMP
- Récupération des informations sur l'image
- Retourne l'image sous forme de texture

Fin ;

---

La deuxième technique dans notre approche que nous avons implémentée est hybridation

L-système billboard. Nous décrivons ci-après la fonction la plus importante que nous avons utilisée pour modéliser les arbres du premiers plan.

Cette fonction permet d'effectuer une interprétation géométrique des symboles constituant le mot généré par la grammaire L-système.

---

Algorithme interprétation géométrique

---

Début

```

Switch (*st)
{
    case '[' : retient la position ;
        break;
    case ']' : revient à la position ;
        break;
    case '+' : rotation autour de l'axe Y ;
        break;
    case 'L' dessin des billboards;
        break;
    case '>' rotation autour de l'axe X ;
        break;
};

```

Fin ;

---

### 3 Résultats Obtenus

Après avoir donné une description de notre système, nous présentons ici les résultats obtenus par notre système pour divers types de scènes, notre modèle peut être appliqué à différents types d'arbre et à des scènes contenant beaucoup d'arbres.

Nous avons mesuré l'impact de notre modèle sur les performances de rendu en rendant un ensemble d'arbres avec différents nombres de quadrilatères. Le système a été développé en langage C++ et la bibliothèque OpenGL .

Dans notre implémentation, nous avons utilisé des images de tailles  $256 \times 256$  (192 Ko).

Nous allons commencer par les arbres représentés par la technique de Billboarding : la figure 63 montre une scène de 1600 arbres, chaque arbre représentée par 8 billboards en croix permettant d'offrir un meilleur effet de perspective et de volume.



**Figure 67.** Une scène de 1600 arbres, chaque arbre représentée par 8 billboards en croix.



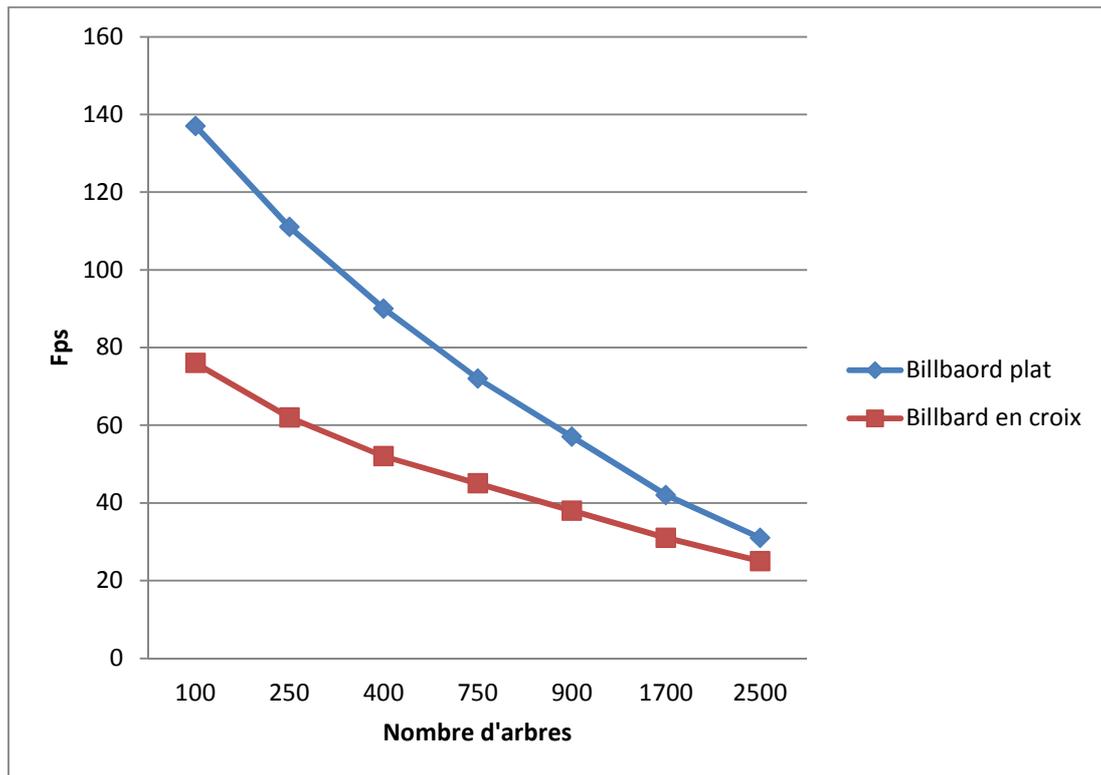
**Figure 68.** *Une scène de 1600 arbres, chaque arbre représentée par 8 billboards en croix.*



**Figure 69.** *Une scène de 3600 arbres chaque arbre représentée par 1 billboard plat.*

Nous présentons quelque résultat démontrant la variation de temps de calcul en fonction du nombre d'arbre et de nombre de billboard. Nous changeant le nombre d'arbre générés ce qui influe directement sur le temps de calcul consommé :

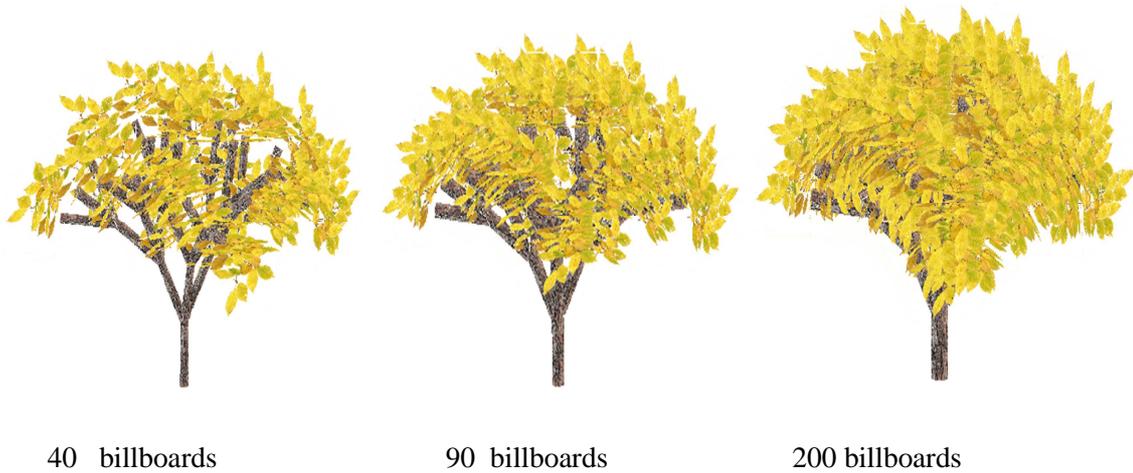
La figure 70 présente une comparaison de fréquence d'affichage de deux scènes construites en utilisant les billboards plat et les billboards en croix.



**Figure 70.** Comparaison entre un modèle Billboard plat et Billboard en croix (8 billboard en croix).

Nous remarquons que pour ces scènes, nous avons réussi à afficher 2500 arbres à une fréquence d'affichage de 25 frames par seconde, ce qui nous ramène à dire que notre rendu est effectué en temps réel.

Dans cette partie nous allons présenter un modèle hybride pour les arbres de premiers plans, la structure de l'arbre est générée par un l-système et les feuilles par les billboards.

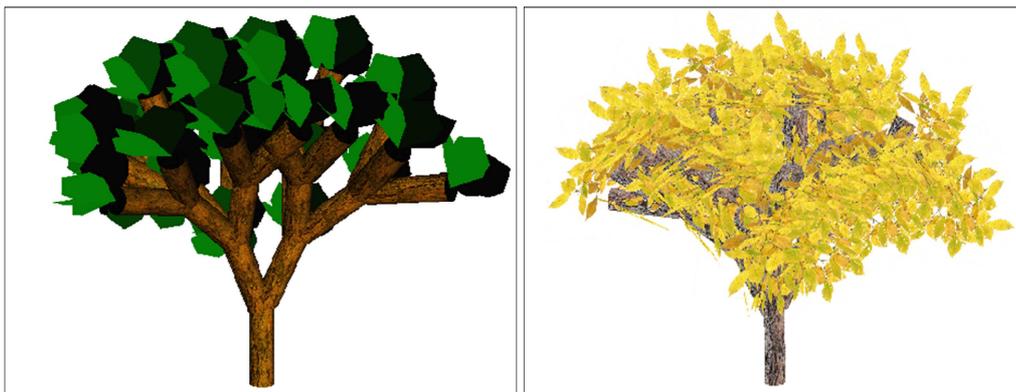


**Figure 71.** *Arbre rendu avec un nombre différent de billboards.*

En vue d'évaluer notre modèle hybride géométrie / imposteur en terme de réalisme visuel, et performance, nous allons faire une comparaison avec un modèle purement géométrique où :

l'écorce est généré par un L-système et les feuilles sont modélisées par un polygone à plusieurs côtés.

La figure 72. présente une comparaison entre un modèle géométrique et un modèle Hybride géométrie / image en terme de qualité visuelle.

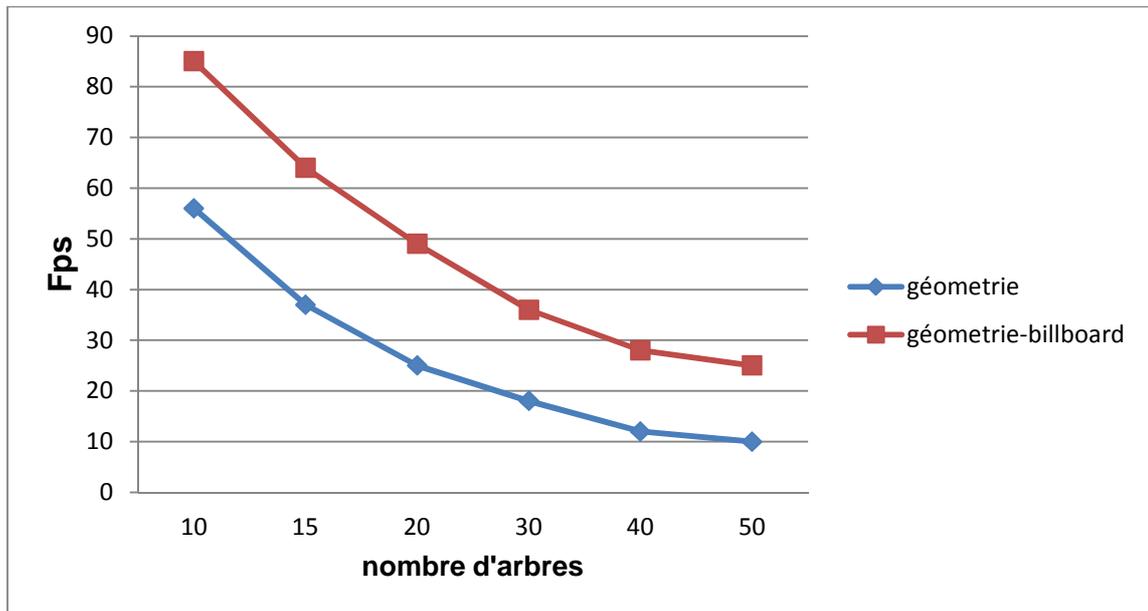




**Figure 72.** *Comparaison entre un modèle géométrique pur et le modèle hybride L-système – Billboard.*

Au niveau de la qualité visuelle, nous avons remarqué que notre méthode augmente le réalisme des arbres avec l'utilisation des billboards qui sont des images réelles de feuilles plaquées sur des quadrilatères.

La figure 73 présente une comparaison de fréquence d'affichage de deux scènes construites en utilisant un modèle géométrique et un modèle Hybride géométrie / image.



**Figure 73.** Comparaison entre un modèle géométrique pur et le modèle hybride  
*Géométrie /image proposé.*

Comme on peut le voir dans le graphique, notre technique qui remplace un ensemble de primitives (feuilles et branches) par une représentation plus efficace (billboard) a permis d'améliorer le temps d'affichage.

Cette technique améliore fortement la complexité et la qualité visuelle présentée dans les scènes complexes de forêts.

Dans cette section nous allons présenter une suite de résultats obtenus à l'aide de notre méthode, dans chaque scène nous changeons le nombre d'arbre et les textures de feuilles et les branches.



**Figure 74.** *Une forêt de 400 arbres.*



**Figure 75.** *Une forêt de 100 arbres.*



**Figure 76.** *Une forêt de 100 arbres avec éclairage et ombrage.*



**Figure 77.** *Une forêt de 520 : 120 arbres billboard-Lsystème, 120 arbres billboard croisé, 280 billboard plats.*

## 4 Discussion

Après avoir présenté les résultats obtenus par notre système, nous pouvons dire que La méthode donne de très bons résultats visuels, ainsi qu'un taux d'affichage très intéressant, le temps de rendu ne dépend plus de la complexité géométrique, il sera essentiellement lié au nombre d'objets composant la scène.

Nous pouvons dire que les performances de notre système est influé par plusieurs paramètre comme :

- le nombre de billboard constituant l'arbre.
- la résolution de texture .
- le nombre d'arbre rendu .

nous nous somme trouver devant un compromis qualité performance . Nous avons tenter de minimiser la complexité en performance avec l'utilisation de deux représentations une représentation 3D où l'observateur peut distinguer des détails sur les arbres de premier plan. La deuxième représentation à l'aide de la technique de billboarding pour les arbres situées à une distance lointaine où l'observateur ne perçoit plus l'arbre mais juste une couverture du relief.

## 5 Limitation du modèle

Permis les limites constatées pour notre modèle :

- Si l'on souhaite représenter une forêt avec des modèles d'arbres différents, on doit stocker autant de textures que l'on souhaiterai obtenir d'arbres visuellement différents.
- l'impossibilité de changer l'illumination ou l'ombrage d'un billboard parce que celle-ci est fixée dans l'image.
- Un effet de discontinuité apparaît aux jonctions des cylindres.

## 6 Conclusion

Les scènes naturelles sont très détaillées et difficiles à modéliser, elles demandent un temps de rendu important et beaucoup de mémoire. Cette complexité provient surtout d'une grande complexité géométrique à toutes les échelles.

Notre système permet de traiter la complexité d'un paysage par l'utilisation des imposteurs (billboarding), afin de diminuer la complexité nous avons utilisé trois type de représentation le choix de représentation est faite suivant la distance qui sépare l'objet est l'observateur (de la version la plus détaillé à la version la moins détaillé).

Le temps de calcul est influé par plusieurs paramètres qui sont :

- le nombre de billboard constituant l'arbre
- la résolution de texture
- le nombre d'arbre rendu

---

## Conclusion et Perspectives

---

Le rendu de scènes naturelles réalistes en temps réel a toujours été un problème difficile à cause de la complexité à toutes les échelles, Cette complexité provient surtout de la géométrie , la grande diversité des éléments naturels ..... , elles demandent un temps de rendu important et beaucoup de mémoire , de nombreux problèmes doivent encore être résolus. La vitesse de rendu est cruciale dans les applications temps réel, mais la qualité d'image est aussi importante.

L'idée directrice développée dans le cadre de cette recherche est d'introduire une représentation qui traite la complexité du foret en remplaçant un ensemble de primitives par une représentation plus efficace.

Cette représentation permettre un rendu d'arbres efficace (qualité) et rapide (temps réel). Nous avons atteint ce but par l'implémentation d'une technique d'imposteurs (billboarding ) en mélangeant différent type de billboard ,et différentes représentations.

Une représentation hybride pour les arbres vue de près, et des billboards avec moins de détail aux arbres situées loin de l'observateur. Ce type de rendu utilise des quadrilatères texturés avec un canal de transparence qui permet de définir précisément la forme du l'arbre , Cependant, les polygones doivent être triés de l'arrière vers l'avant, ce qui est une opération coûteuse.

Nous constatons que cette technique améliore fortement les performances et la qualité visuelle présentes dans les scènes complexes de forêts.

Finalement, nous pouvons conclure que :

- chaque scène naturelle possède un type de complexité spécifique ce qui implique un traitement spécifique à cette complexité.
- Il n'existe pas d'une méthode universelle pour traiter la complexité .

- L'utilisation de niveau de détail pour les scènes naturelles est une nécessité afin de diminuer la complexité.

Pour le travail futur, Comme dans toutes les applications très consommatrices de calculs , il est tout à fait possible d'envisager l'utilisation de la puissance de calcul des cartes graphiques actuelles (GPUs) et la parallélisation de cet algorithme pour des raisons de performance optimale et pour laisser le processeur central (CPU) libre d'effectuer d'autres tâches.

Dans la nature, les arbres sont souvent influencés par le vent. Pour simuler ce mouvement des arbres, nous déformons le billboard dans leurs quatre coins, la dernière proposition est d'insérer d'autres objets naturels come : l'herbe, les roches ...

---

## Bibliographie

---

- [AB91] E. H. Adelson and J. R. Bergen, The Plenoptic Function and the Elements of Early Vision, Computational Models of Visual Processing, Edited by Michael Landy and J. Anthony Movshon, The MIT Press, 1991.
- [AK84] M. Aono and T. L. Kunii. Botanical tree image generation. IEEE Computer Graphics and Applications, 4(5):10-34, May 1984.
- [Cat74] E. E. Catmull. A Subdivision Algorithm for Computer Display of Curved Surfaces. Phd. thesis, University of Utah, December, pages 42, 51, 1974.
- [Cc97] C. Chaudy. Modélisation et rendu d'images réalistes de paysages naturels, Thèse de doctorat soutenue à Grenoble, 1997.
- [Ch95] S. Chen, QuickTime VR, An Image-Based Approach to Virtual Environment Navigation, Computer Graphics (SIGGRAPH'95), pp. 29-38, August 1995.
- [CW93] Chen, S. E. et Williams, L, View Interpolation for Image Synthesis, Computer Graphics (SIGGRAPH'93 Proceedings), pp. 279–288. 1993
- [DCV97] Lucia Darsa, Bruno Costa Silva, and Amitabh Varshney. Navigating static environments using image-space simplification and morphing. In Proceedings of the 1997 symposium on Interactive 3D graphics, pages 25, 1997.
- [DDSD03] Xavier Décoret, Frédo Durand, François X. Sillion, and Julie Dorsey. Billboard clouds for extreme model simplification, ACM Transactions on Graphics, pp. 689–696, July 2003.

- [DEC02] Xavier Décoret, Frédo Durand, François X. Sillion et Julie Dorsey, Billboard Clouds , Rapport d'activité INRIA, juin 2002.
- [DF09] Philippe Decaudin and Fabrice Neyret, Volumetric Billboards , Eurographics Symposium on Rendering, France, 2009
- [DREF88] P. de Reffye, C. Edelin, J. Françon, M. Jaeger, and C. Puech. Plant models faithful to botanical structure and development. In SIGGRAPH 1988, Computer Graphics Proceedings, pages 151–158, August 1988.
- [JWS02] Stefan Jeschke, Michael Wimmer, and Heidrun Schumann. Layered environment map impostors for arbitrary scenes. In Wolfgang Sturzlinger and Michael McCool, editors, Proceedings of Graphics Interface, pages 1-8, May 2002.
- [HAA97] Youichi Horry, Ken-Ichi Anjyo and Kiyoshi Arai, Tour into the picture: using a spidery mesh interface to make animation from a single image, SIGGRAPH'97, pages 225-232, 1997.
- [HQU00] Huamin Qu, Sample-based Rendering, Rapport d'activité, Department of Computer Science, State University of New York at Stony Brook, 2000.
- [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, 331-338, 1971.
- [GOR96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski et Michael F. Cohen, The Lumigraph, Rapport d'activité, SIGGRAPH, 1996.
- [Gre89] Ned Greene. Voxel space automata: Modeling with stochastic growth processes in voxel space. In Jerrey Lane, editor, Computer Graphics (SIGGRAPH '89 Proceedings), volume 23, pages 175-184, July 1989.
- [LH96] Marc Levoy and Pat Hanrahan. Light field rendering. In Proceedings of the ACM Conference on Computer Graphics, pages 31.42, New Orleans, LA, USA, August 1996.

- [Lin68] Lindenmayer. Mathematical models for cellular interaction in development. Journal of Theoretical Biology, page 280–315, 1968.
- [MB95] Leonard McMillan and Gary Bishop. Plenoptic modeling : An image-based rendering system. Proceedings of SIGGRAPH 95, California, pages 39.46, August 1995.
- [Mc97] Leonard McMillan. An Image-Based Approach to Three-Dimensional Computer Graphics, PhD thesis, Department of Computer Science, University of North Carolina, April 1997.
- [MEY01] Alexandre Meyer, Représentations d’arbres réalistes et efficaces pour la synthèse d’images de paysages, Thèse de doctorat, Université Joseph Fourier, France, 10 décembre 2001.
- [MS95] Paulo W. C. Maciel and Peter Shirley. Visual navigation of large environments using textured clusters. In ACM Siggraph Symposium on Interactive 3D Graphics, 1995.
- [OGL98] Advanced Graphics Programming Techniques Using OpenGL, ACM SIGGRAPH Course Notes, 1998.
- [OK93] M. Okutomi, T. Kanade, A Multiple-Baseline Stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pages 353-363, April 1993.
- [PC01] Frank Perbet and Marie-Paule Cani, Animating Prairies in Real-Time, Proceedings of the Conference on the 2001 Symposium on interactive 3D, Graphics, 2001.
- [Pel04] Kurt Pelzer. GPU Gems, chapter 7 - Rendering Countless Blades of Waving Grass, pp. 107-121. Addison-Wesley, March 2004.
- [PL90] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. The algorithmic beauty of plants, Springer-Verlag, New York, NY, USA, 1990.
- [POR04] Damien PORQUET, Rendu en temps réel de scènes complexes, Thèse de doctorat, université de Limoges, 29 Novembre 2004.

- [Kaw82] K. Kawaguchi. A morphological study of the form of nature. In *Computer Graphics (SIGGRAPH '82 Proceedings)*, volume 16, pages 223-232, July 1982.
- [KS05] Kimura K et Saito, H, Video Synthesis at Tennis Player Viewpoint from Multiple View Videos , *IEEE Virtual Reality (VR)*, Bonn, Germany, pp. 281–282, 2005.
- [RB85] W. T. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In *SIGGRAPH 1985, Computer Graphics Proceedings*, pages 313–322, July 1985.
- [Ree83] W.T. Reeves. Particle systems a technique for modeling a class of fuzzy objects. *ACM Trans. Graphics*, pages 91–108, April 1983.
- [RNK97] P. Rander, P.J. Narayanan, and T. Kanade, Virtualized reality: constructing time-varying virtual worlds from real world events, *Proc. of IEEE Visualization*, pages 277-283, October 1997.
- [Sch95] Gernot Schaufler. Dynamically generated impostors. *GI Workshop on Modeling, Virtual Worlds, Distributed Graphics*, Bonn, Germany, pages 129.139, 1995.
- [Sch97] Gernot Schaufler. Nailboards : A Rendering Primitive for Image Caching in Dynamic Scenes. In *Proc. 8th Eurographics Workshop on Rendering*, pages 151.162. Springer Verlag, 1997.
- [Sch98] Gernot Schaufler. Image-based object representation by layered impostors. In *EurographicsWorkshop on Rendering 98*, pages 145–156, 1998.
- [SCG97] Sloan, P. P. , Cohen, M. F. et Gortler, S. J, Time critical Lumigraph rendering, *Symposium on Interactive 3D Graphics*, Providence, USA, pp. 17–23, 1997.
- [SD96] Steven M. Seitz and Charles R. Dyer. View morphing. In *Proceedings of the ACM Conference on Computer Graphics*, New Orleans, USA, pages 21.30, August 1996.
- [SD97] S. M. Seitz and C. R. Dyer, Photorealistic Scene Reconstruction by Voxel Coloring, in *proc, Computer Vision and Pattern Recognition Conference*, pages 1067-1073, 1997.

- [SH99] Heung-Yeung Shum and Li-Wei He. Rendering with concentric mosaics, In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 299-306,1999.
- [SRC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 43.54. ACM Press, 1996.
- [TSK01] Hai Tao, Harpreet S. Sawhney, and Rakesh Kumar, A global matching framework for stereo computation, in proc. IEEE International Conference on Computer Vision, ICCV'01, pages 532-539, 2001.
- [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.
- [Vn06] M Vincent Nozick , Video-Based Rendering and Virtual Reality, Thèse de doctorat, Université de Marne-La-Vallée, 7 juin 2006.
- [Wha05] David Whatley. GPU Gems 2, chapter 1 - Toward , Photorealism in Virtual Botany, pp. 7-25. Addison-Wesley, March 2005.
- [WSK02] Bennett Wilburn,Michael Smulski, Hsiao-Heng Kelin Lee and Mark Horowitz, The Light Field Video Camera, in proc. of Media Processors, SPIE Electronic Imaging 2002, pages 29-32, 2002.
- [YO01] Yasemin Kuzu, Olaf Sinram. ‘ Phoptorealistic object reconstruction using voxel coloring and adjusted image orientations ‘.Photogrammetry and Cartography Technical University of Berlin Germany 2001.
- [YSK+02] Shuntaro Yamazaki, Ryusuke Sagawa, Hiroshi Kawasaki, Katsushi Ikeuchi, and Masao Sakauchi. Microfacet billboarding. In Proceedings of the 13th Eurographics workshop on Rendering, pages 169.180. Eurographics Association, 2002.

- [ZK01] Ye Zhang, Chandra Kambhampati, On 3D Scene Flow and Structure Estimation, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai Marriott, pages 778-785, 2001.

