

UNIVERSITÉ DE LIMOGES
ÉCOLE DOCTORALE « Sciences et Ingénierie pour l'Information »
FACULTÉ DES SCIENCES ET TECHNIQUES
Laboratoire DMI), équipe SIR

Thèse N° 42-2011

THÈSE
pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ DE LIMOGES
Discipline / Spécialité : Informatique et Applications
présentée et soutenue par
Alexandre Peyrat
le 07 octobre 2011

Modélisation, simulation et visualisation de détails dans les scènes naturelles réalistes

*Thèse dirigée par Djamchid Ghazanfarpour,
Co-encadrée par Stéphane Mérillou et Olivier Terraz*

JURY

Rapporteurs :

Mme. Dominique BECHMANN Professeure à l'Université de Strasbourg

M. Mateu SBERT Professeur à l'Université de Girona

Examineurs :

M. Djamchid GHAZANFARPOUR Professeur à l'Université de Limoges

M. Pascal LIENHARDT Professeur à l'Université de Poitiers

M. Stéphane MERILLOU Maître de conférences HDR à l'Université de Limoges

Invité :

M. Olivier TERRAZ Maître de conférences à l'Université de Limoges

*À Oana, qui rend ma vie plus belle,
À ma famille.*

Remerciements

Je tiens tout d'abord à remercier madame Dominique Bechmann et monsieur Mateu Sbert ainsi que monsieur Pascal Lienhardt d'avoir accepté d'examiner mes travaux.

Je remercie Djamchid Ghazanfarpour, mon directeur de thèse, de la liberté et la confiance qu'il a su m'accorder. Merci également à Stéphane Mérillou et Olivier Terraz d'avoir pu encadrer mes travaux malgré un emploi du temps chargé. Le temps passé à travailler avec vous fut un réel plaisir et les échanges furent aussi enrichissants qu'intéressants.

Merci à mes collègues, Cyril, Richard, Emmanuelle et tous les autres. Je remercie particulièrement Mickaël de son amitié et pour nos expériences entrepreneuriales ainsi que Hubert pour son écoute et son amitié.

Merci aux anciens de Jidé, Suzy, Stan, Koso, Cyrion, Geekette et la presque ancienne Sylvie ainsi que ceux que j'oublie pour m'avoir initié à la "recherche".

Je remercie mes parents et ma soeur d'avoir été là et de m'avoir supporté. Leur soutien et leur compréhension ont toujours été sans faille.

Merci à mes amis pour leur bonne humeur et leur soutien. Un merci tout particulier à Pierre-Jean pour me comprendre et pour nos discussions de geek au coin du feu.

Enfin, merci à toi, Oana pour ton soutien et ta présence à mes côtés. Merci pour cette année passée avec toi et pour toutes celles à venir..

Résumé

Le réalisme d'une scène naturelle est fortement dépendant de la quantité et la variété d'objets la composant. Ce nombre d'éléments ainsi que leurs différences au sein d'une même famille constituent une hétérogénéité nécessaire de ce réalisme. Le vieillissement est un autre aspect important pour l'étude des scènes naturelles. En effet, une scène sans défauts paraîtrait trop "neuve" et peu réaliste.

La modélisation et le rendu de ces détails posent des problèmes couvrant plusieurs domaines de recherche. Nous devons fournir, pour chaque famille d'objets désirée, la possibilité d'en générer un grand nombre différent les uns des autres tout en les conservant identifiables. Ces objets doivent aussi pouvoir subir des formes de vieillissement afin de s'intégrer au mieux dans la scène naturelle désirée. Ces vieillissements spécifiques se traduiront par une altération des formes, des couleurs et de l'aspect des éléments qui peupleront la scène.

Premièrement, pour explorer les détails à une petite échelle, nous nous sommes intéressés à la modélisation et au rendu de feuilles qui sont particulièrement présentes dans les scènes naturelles. Elles possèdent une vaste gamme de formes, de couleurs et de processus de vieillissement ce qui en fait un très bon exemple de variété. Notre modèle permet de générer le feuillage entier de végétaux à n'importe quel âge. Les feuilles générées seront toutes différentes dans une même famille, leurs formes et leurs couleurs différeront tout en restant semblables.

La seconde partie de nos travaux porte sur la création de détails à plus grande échelle. L'altération du sol nous permet d'explorer ce niveau d'échelle. Quel que soit le matériau sur lequel on se trouve, les interactions avec le sol vont créer des traces. Dans les scènes naturelles, les piétons, les véhicules ou les animaux altèrent les sols où ils évoluent. Notre modèle se compose d'une trajectoire et des caractéristiques des empreintes. Le sol est ensuite altéré pour former des traces et des dépôts de matière le long de la trajectoire.

Mots clefs : Synthèse d'images, rendu réaliste, phénomène naturel, phénoménologie, vieillissement, feuille, empreinte, sol.

Abstract

The realism of a natural scenery is heavily dependent on the amount and the variety of objects composing it. This number of elements and their differences within a same family contribute to the necessary heterogeneity. Aging is another important aspect ; a scenery without flaws would appear too “new“ and unrealistic.

Modeling and rendering of these details raise several problems covering several research fields. We must provide for the desired class of elements the ability to generate many distinct but recognizable items. These objects should also be aged in order to better fit into the desired natural scenery. This aging will result in a change of shape and color of the considered elements.

Our first interest was directed towards a little scale of details. For that purpose, we studied the modeling and rendering of leaves, which are especially present in natural sceneries. They have a wide range of shapes and colors, and undergo different aging processes which makes leave a good exemple of variety. Our model is able to generate the full foliage of plants at any age. The resulted leaves will all be different from each other for the same family ; their shapes and colors will vary while remaining similar.

The second part of our work involves details at an upper scale. The alteration of soil grant us the ability to explore this field of details. Whatever the material composing the ground, the different interactions with it will create traces. In natural sceneries, the soil is affected by pedestrians, vehicles or animals. Our model consists of a trajectory and the characteristics of fingerprints. The soil is altered to form traces. Moreover, material deposits are considered along the trajectory.

Key words : Computer graphics, realistic rendering, natural phenomena, phenomenology, aging, leaf, imprint, ground.

Table des matières

Introduction	1
1 Etat de l'art.	5
1.1 Eléments fondamentaux	8
1.1.1 Cartes Généralisées	8
1.1.1.1 Description du modèle	8
1.1.1.2 Outils des Cartes Généralisées	9
1.1.1.3 Autres modèles de représentation topologique	10
1.1.1.4 Conclusion	14
1.1.2 Les systèmes de Lindenmayer (L-Systèmes)	14
1.1.2.1 Le D0 L-Système	15
1.1.2.2 Interprétation graphique des L-Systèmes	15
1.1.2.3 Le L-Système paramétrique	17
1.1.2.4 L-Systèmes dépendants du contexte	18
1.1.2.5 L-Système stochastique.	19
1.1.2.6 Map L-Système.	20
1.1.2.7 Conclusion	21
1.2 Génération de petits détails : le choix des feuilles	21
1.2.1 Génération de feuilles	21
1.2.1.1 Modélisation à partir de L-Systèmes	21
1.2.1.2 Modélisation basée image.	24
1.2.1.3 Algorithme biologique.	26
1.2.1.4 Modélisation par squelette.	28
1.2.1.5 Surfaces implicites.	30
1.2.1.6 Discrete Shells.	32
1.2.1.7 Système de particules.	32
1.2.1.8 Conclusion	33
1.2.2 Rendu de feuilles	34

1.2.2.1	Rendu sous surfacique.	34
1.2.2.2	Génération de texture avec un système de particules.	39
1.2.2.3	Evolution et vieillissement.	41
1.2.2.4	Conclusion	41
1.3	A une échelle supérieure : le choix des altérations de terrains	42
1.3.1	Altérations de terrains	42
1.3.1.1	Érosion	42
1.3.1.2	Fissures	47
1.3.1.2.1	Les modèles géométriques	48
1.3.1.2.2	Les modèles physiques	48
1.3.1.3	Altération du sol par des phénomènes extérieurs	49
1.3.1.3.1	Modèles physiques	50
1.3.1.3.2	Modèles empiriques	54
2	Génération de grandes quantités de feuilles à différents âges	57
2.1	Introduction	59
2.2	Problématique	60
2.3	Le 2-G-Carte L-Systeme paramétrique	61
2.3.1	Notations et outils	61
2.3.2	Génération de la forme de la feuille	62
2.3.2.1	Définition des règles de construction	62
2.3.2.2	Amélioration de la grammaire	67
2.3.2.3	Création de grande quantité de feuilles	75
2.4	Vieillissement de la feuille	80
2.4.1	Aspects géométriques du vieillissement	80
2.4.2	Vieillissement colorimétrique	83
2.5	Résultats	85
2.6	Conclusion	92
3	Génération d'empreintes et de traces sur des terrains	95
3.1	Introduction	97
3.2	problématique	98
3.3	Un modèle topologique	99
3.4	Génération de trajectoires	104
3.5	Enfoncement de la trajectoire et dépôt de matière	108
3.6	Résultats	117
3.6.1	Extensions	122

3.7 Conclusion	124
Conclusion et perspectives	125
Annexes	129
Annexe	128
Bibliographie	137

Table des figures

1.1	La représentation topologique d'une 2-G-Carte avec ses différentes involutions.	9
1.2	Nous pouvons voir sur la gauche la cellule sommet, au centre la cellule arête et sur la droite la cellule face.	10
1.3	Flocon de neige de VON KOCH. De gauche à droite et de haut en bas nous pouvons observer le résultat pour 0, 1, 2 et 3 itérations . .	17
1.4	L-Système paramétrique.	18
1.5	Propagation depuis la racine des segments "I" dans un L-Système dépendant du contexte.	19
1.6	Trois générations avec le même L-Système stochastique.	20
1.7	Map L-Système simple.	20
1.8	Un exemple de cylindre généralisé.	22
1.9	Conversion des fonctions tag en paramètres.	23
1.10	Divers squelettes générés avec les fonctions tag.	23
1.11	Rendu des résultats de la méthode de [RLFS02]	24
1.12	Pipeline de la méthode utilisée dans [QTZ ⁺ 06] pour générer un végétal.	24
1.13	résultat de la modélisation basée image de [QTZ ⁺ 06].	25
1.14	Les différentes phases de la segmentation des feuilles.	26
1.15	Etape initiale de l'algorithme.	27
1.16	Exemple de fonctionnement de la méthode de [RFL ⁺ 05].	27
1.17	A droite nous pouvons voir la photo d'une feuille et à gauche la même feuille générée par [RFL ⁺ 05].	28
1.18	Silhouette et squelette de la feuille lobée.	29
1.19	Les différentes étapes de la création du maillage.	29
1.20	Maillage et rendu résultats de la méthode de modélisation par squelette.	30
1.21	Développement d'une feuille en contours implicites.	30
1.22	Différents rayons d'influence interpolés.	31
1.23	Méthode de traçage du contour implicite.	31

1.24	Modélisation avec les Discrete Shells.	32
1.25	Transport rectiligne des particules.	32
1.26	Transport dépendant des autres particules.	33
1.27	Développement du réseau de nervures.	33
1.28	Coupe transversale d'une feuille de hêtre.	34
1.29	reflectance et transmittance de la lumière sur une feuille de hêtre.	35
1.30	Les différentes couches utilisées dans le modèle de [BR97] et un trajet possible d'un rayon de lumière.	36
1.31	Perturbations du rayon propagé.	37
1.32	Résultats du rendu avec le FSM. Une source de lumière éclaire les feuilles par l'arrière ce qui laisse apparaître le réseau de nervures.	38
1.33	Modèle en tranche d'une feuille.	38
1.34	Résultats de la méthode de [WWD ⁺ 05].	39
1.35	Modification du champ de vecteurs avec le mouvement des particules.	40
1.36	Dispersion des couleurs dans la texture.	40
1.37	Résultats du rendu de [YRL04] avec une plante grimpante	40
1.38	Modification du relief au cours des itérations ([CMF98]).	43
1.39	Le modèle en couche de Benes ([BF01]).	44
1.40	L'érosion des différentes couches est indépendante. Ici le "W" est fait d'une matière très dure et est recouvert de boue.	44
1.41	L'érosion d'une montagne réelle, l'eau courante apparaissent en noir.	45
1.42	Dissolution du sol par l'eau.	45
1.43	Transport de l'eau et des sédiments dans les éléments du modèle (ici avec deux éléments). L'instant $t - 1$ se trouve à gauche et l'instant t à droite.	46
1.44	Érosion et infiltration dans un sol érodé par la pluie.	47
1.45	Génération d'arches et de falaises avec une érosion créant des amas de rochers aux pieds des conformations rocheuses.	47
1.46	Utilisation de la tessellation de Dirichlet sur plusieurs niveaux dans [PMRDM96]	48
1.47	Résultats de la méthode de Valette <i>et al.</i> [VPLL08].	49
1.48	Fissures en utilisant la mécanique du solide et les éléments finis	49
1.49	Le sol est modifié par la lame d'un bulldozer avec la méthode de Li <i>et al.</i>	51
1.50	Un piéton laisse une empreinte de chaussure dans la neige grâce à la méthode de Sumner <i>et al.</i>	52

1.51	Les différents masques fournissent les niveaux de détails du terrain (sur la gauche) et permettent d'obtenir le rendu final (sur la droite).	53
1.52	La chute d'un Rubik's cube (de gauche à droite) laisse sur le sol une empreinte en forme de carré.	54
1.53	Les ornières réalisées dans [ABZO06] en temps réel par un traitement sur GPU.	55
1.54	Les traces de roues visualisées avec des niveaux de détails par [CZXY07]. Sur la gauche, sans zones de transition et sur la droite avec.	55
2.1	Le principe de fonctionnement de notre modèle de génération de feuilles.	60
2.2	Une face B est collée par son arête origine sur les arêtes extrémité libres des faces A	64
2.3	La croissance du modèle à partir d'une face A . nous pouvons observer l'état initial puis les itérations 1 et 3.	64
2.4	L'itération numéro 3 de la grammaire 2.2 visualisée grâce à notre modèle. Les faces jaunes ont pour label A , les vertes B	65
2.5	Le collage de deux arêtes adjacentes. Ici quand on trouve une arête A_{C_1} adjacente à une arête B_{C_2} , ces deux arêtes sont collées.	65
2.6	Le résultat de la première itération de la grammaire 2.3. Deux faces B sont ajoutés sur les arêtes A_{C_1} et A_E puis les arêtes B_{C_1} et B_{C_2} sont collées.	66
2.7	La découpe d'une face A en deux faces B et C suivant l'arête A_O . Cette figure nous montre le résultat de la grammaire 2.4.	67
2.8	Surcharge des faces A . Nous pouvons voir ici le résultat de la grammaire 2.7 pour 6 itérations.	70
2.9	Les résultats de la grammaire 2.8 pour des angles α valant 90, 45 et 25 degrés de gauche à droite.	71
2.10	La création de nervures secondaires courbées grâce à la surcharge des paramètres. Chaque face est ajoutée lors de la croissance des nervures secondaires avec un angle différent de son angle d'origine.	71
2.11	Ici la nervure principale se divise en deux à mi-génération grâce à une règle conditionnelle.	73
2.12	Le remplissage au cours des itérations. Les faces rouges sont les faces du limbe ajoutées à chaque itération à la suite la génération des nervures.	74
2.13	De gauche à droite : la variable est fixée à zéro, pour ne garder que le réseau de nervures puis à cinq pour ne garder que la moitié du limbe et enfin à dix pour avoir tout le corps de la feuille.	75

2.14	Gauche : le résultat de la génération de la grammaire 2.10 pour 8 itérations. Droite : l'introduction d'angles aléatoires dans la grammaire 2.10. La feuille paraît plus réaliste, car moins "parfaite". . . .	77
2.15	Ici, un des résultats donnés par la grammaire 2.11 avec l'ajout de paramètres aléatoires. Nous pouvons voir les tailles différentes pour les faces de la nervure principale.	78
2.16	Les différentes feuilles de tilleul générées. La longueur des nervures augmente de gauche à droite et les valeurs des angles de haut en bas.	78
2.17	Ces 20 feuilles sont générées à partir de la même grammaire. Les différences inter-espèce sont dues à l'ajout de variables aléatoires pour les angles des nervures.	79
2.18	Nous pouvons observer sur cette figure cinq niveaux de détail pour une feuille de trèfle. De gauche à droite nous voyons les itérations 5, 6, 7 et 8 de la grammaire pour créer ce trèfle à quatre feuilles. Le nombre de faces utilisées augmente de gauche à droite en suivant le nombre d'itérations.	79
2.19	De gauche à droite nous pouvons voir la croissance d'une feuille de châtaignier de la sortie du bourgeon à sa taille adulte.	81
2.20	Le cycle complet du vieillissement géométrique d'une feuille de châtaignier. De la gauche pour la naissance à la droite pour la fin de vie, nous pouvons observer les différentes étapes du vieillissement de cette feuille.	81
2.21	Deux dégradations possible dues à des éléments externes. Ici la pointe de la feuille de gauche s'est vue amputée, la feuille de droite a subi une déchirure.	82
2.22	Sur la gauche, une attaque de chenille sur une feuille réelle. Sur la droite nous pouvons voir la génération des attaques d'insectes, les parties rouges sont les morceaux de feuille manquants.	83
2.23	Les couleurs d'une feuille de tilleul à différents âges. Nous avons utilisé le paramètre de mélange pour simuler le vieillissement de la feuille.	84
2.24	L'action du masque de mélange sur la génération de la texture d'une feuille de lierre. Nous pouvons observer sur la gauche la couleur jeune, le masque de mélange puis la couleur vieille. Sur la droite la feuille texturée en utilisant le masque et les deux couleurs.	85
2.25	Différentes formes de feuilles (schémas) ainsi que leurs représentations obtenues avec notre modèle.	86

2.26	A gauche, deux feuilles de liquidambar ayant subit des vieillissements différents. Pour cela nous avons utilisé des masques de mélange différents en gardant les couleurs choisies par l'utilisateur. A Droite des feuilles réelles ayant le même type de coloration	87
2.27	Deux scènes similaires peuplées avec à gauche, uniquement des feuilles comportant beaucoup de détails et à droite des feuilles de moins en moins détaillées suivant l'éloignement sur la route. Les deux scènes semblent identiques.	87
2.28	Quatres espèces de feuilles modélisées avec notre méthode. Nous pouvons voir les photographies de feuilles servant de modèle, la génération de la grammaire et des textures puis enfin le rendu final de la feuille.	88
2.29	A gauche une tige de rosier parfaite comportant sept feuilles identiques. A droite chaque feuille de cette branche est différente ce qui apporte beaucoup de réalisme à la plante.	89
2.30	L'illustration des différentes étapes du vieillissement d'une tige de rosier.	89
2.31	Différents niveaux d'attaques d'insectes. A droite seules les nervures restent là ou les insectes ont consommé le limbe de la feuille. . . .	90
2.32	Une branche de vieux lierre grim pant sur une poutre.	90
2.33	Quatre saisons différentes pour un arbre et un plan de lierre. Le printemps, la fin de l'été, l'automne et l'hiver sont présent sur cette figure.	91
2.34	L'utilisation de feuilles différentes impacte le réalisme de la scène. Ici 40 puis 2 feuilles différentes sont utilisées pour peupler la scène (de gauche à droite). Plus le nombre de feuilles différentes faible, plus le réalisme de la scène l'est aussi.	91
3.1	Les données fournies à notre modèle sont utilisées puis des outils sont générés pour effectuer le rendu du terrain altéré.	99
3.2	Comparaison des temps de recherche de voisins. En bleu, notre modèle basé sur une 2-G-Carte. En orange, un maillage classique. .	100
3.3	Les différentes involutions mises en jeux dans le modèle des 2-G-Cartes. Ici nous obtenons deux triangles liés par une arête centrale.	101
3.4	Les liaisons β_1 nous permettent de lier la structure du sol sous forme de couches à chaque brin de la 2-G-Carte.	101
3.5	Deux triangles sont subdivisés chacun en quatre triangles suivant le principe du PTQ. Nous pouvons observer sur cette figure en jaune les liaisons α_0 , en rouge les α_1 et en vert les α_2	102

3.6	Création et affectation de nouvelles couches pour les brins créés par raffinement d'un segment. En haut le segment initial avec à gauche une couche de boue surplombant une couche de roche et à droite une couche de roche à nue. En bas le segment après raffinement et création des nouvelles couches.	103
3.7	Trois trajectoires dessinées par l'utilisateur. Nous pouvons observer les changements induits par la modification des paramètres.	104
3.8	Différents types d'empreintes avec leurs cartes de hauteurs associées.	105
3.9	Des textures d'empreintes nous permettant de simuler d'autres comportements de marche ou de course.	106
3.10	L'impression d'une trajectoire sur le maillage. Nous pouvons voir en haut à gauche l'empreinte utilisée pour générer la trajectoire.	107
3.11	Seul le bord de l'empreinte se voit ajouter des détails. Le centre change peu de résolution.	107
3.12	A gauche, la vue de la caméra sur des traces de véhicules. A droite, les mêmes traces vues perpendiculairement. Nous voyons bien la gradation des détails en fonction de l'éloignement à la caméra.	108
3.13	Enfoncement d'une empreinte de pieds en fonction du poids. Plus le poids augmente plus l'enfoncement est conséquent. Nous pouvons voir une vignette en haut à gauche de chaque image représentant la carte de hauteur utilisée pour l'empreinte de pied.	109
3.14	Extrusion topologique des bords d'une empreinte dans notre modèle. Ici la face A est B sont reliées par un rebord constitué des faces F1 et F2.	110
3.15	La carte de hauteur de l'empreinte est plus ou moins altérée en fonction de la vitesse. Ici la vitesse du sujet augmente de gauche à droite.	111
3.16	A gauche, un pneu comportant plusieurs motifs se répétant sur sa surface. Nous pouvons voir sur la droite l'empreinte déduite ainsi que sa carte de hauteur.	112
3.17	Enfoncement d'une empreinte en tenant compte des couches de matériaux. En haut la structure du sol intacte, en bas la même structure altérée par une empreinte plate.	113
3.18	Enfoncement d'une empreinte sur le bord d'un monticule de terre. Nous pouvons observer l'enfoncement sélectif qui s'opère sur la terre et non sur le béton soutenant le monticule.	114
3.19	Ici, trois tampons simples sont utilisés pour gérer les dépôts de matière. Nous pouvons voir en haut à gauche les trois tampons générés et en rouge les endroits où la matière est déposée ainsi qu'un lien vers le tampon responsable du dépôt.	115

3.20	La texture générée après le passage d'un marcheur dans une flaque de boue. Nous pouvons observer en blanc les zones où il y a de la matière, c'est à dire la flaque ainsi que les dépôts de matière le long de la trajectoire après la flaque. Les zones noires représentent le matériau de base du terrain qui peut être de la roche dans notre cas.	117
3.21	Des empreintes de pieds dans un sable de bord de mer en haut et dans de la boue en bas.	117
3.22	A gauche, une trajectoire de marques de pieds dans le sable. A droite, Un marcheur dans le sable vient de faire demi-tour.	118
3.23	Des centaines d'empreintes sur le sol, nous pouvons voir trois trajectoires se croisant avec un grand nombre d'empreintes sur chacune d'elle. Un tel résultat ne peut être obtenu sans traitement automatique qu'avec beaucoup de temps.	118
3.24	Nous pouvons observer sur cette figure plusieurs empreintes de mouettes suivant plusieurs trajectoires. En bas à gauche figurent l'empreinte utilisée ainsi que la carte de hauteur correspondante. Des zooms sont effectués sur des trajectoires différentes pour montrer le résultat de leur génération.	119
3.25	A gauche : de la matière déposée par un véhicule. A Droite : un marcheur dépose du sable sur le bitume. Nous pouvons voir sur chaque image les formes d'empreintes utilisées avec en haut la forme générale de l'empreinte et en bas la carte de hauteur correspondante.	119
3.26	Nous pouvons voir sur cette figure les empreintes laissées par un cheval venant de traverser une route.	120
3.27	Nous pouvons voir sur cette figure les empreintes laissées par un cheval au galop sur un chemin terreux de campagne.	120
3.28	Simulation d'une trajectoire de véhicule.	121
3.29	Passage d'engins sur une route. Ces engins entrent sur la route par différents endroits et y laissent des marques conséquentes.	121
3.30	A gauche une photo de l'écorce d'un arbre. A droite la texture représentant les zones de surface (noires) et les creux (blancs).	122
3.31	Nous pouvons observer sur la gauche le maillage avec les zones de surface de l'écorce et à gauche la détection des bords de ces zones .	123
3.32	Exemple de génération du maillage d'une écorce très rugueuse. . .	123

Introduction

La plupart des oeuvres artistiques, photographiques et cinématographiques montrent des scènes naturelles. Pour synthétiser de telles scènes, le réalisme est primordial et passe par un grand nombre de détails. Nous pouvons citer, par exemple, les restes végétaux situés sur le sol tels que les brindilles, les morceaux d'écorces, les épines de pins ou les feuilles (vertes ou mortes) tombées au sol. Les altérations du milieu naturel entrent aussi dans ces familles de détails. Il y a par exemple les fissures présentes sur de la roche, les traces laissées sur le sol après le passage d'une personne ou les altérations subies lors de l'érosion du terrain.

Les scènes naturelles doivent leur réalisme en grande partie à la diversité des objets qui les composent. Cette diversité se traduit aussi bien par une large variété d'éléments différents que par des éléments similaires au sein d'une même famille. Cette hétérogénéité est nécessaire au réalisme d'une scène naturelle, car elle apporte l'aspect à la fois naturel et désordonné que l'on rencontre dans la nature. En effet, tous les éléments appartenant à une même catégorie de détails ne sont pas similaires et pourtant subissent le même type de développement. Quantité de phénomènes extérieurs influent sur la formation et la dispersion des éléments pour une famille d'objets donnée. L'oeil humain est capable de faire le tri dans ces différents éléments pour les découper en plusieurs classes. Il ne remarque pas forcément les différences intrinsèques à chaque famille, mais les identifie aisément. Le fait est que le réalisme d'une scène provient en partie de ces différences.

Le vieillissement des scènes naturelles est aussi un facteur de réalisme important. Une scène totalement propre et "neuve" laisse une impression de perfection irréaliste. De ce fait, les détails nous intéressant doivent posséder plusieurs niveaux de vieillissement distincts présents dans la scène souhaitée. Une scène hivernale n'admettra pas de feuilles vertes sur les arbres et à l'inverse une scène estivale ne pourra pas contenir une grande quantité de feuilles mortes sur le sol. De plus, le vieillissement de certains éléments se fait aussi bien sur la forme que sur la couleur. Une pomme va changer de couleur avec l'âge, mais va aussi se flétrir et sécher.

Problématique

La modélisation et le rendu de détails dans les scènes réalistes comportent différents aspects. Tout d'abord la modélisation proprement dite de ces détails, il faut pouvoir modéliser précisément chaque classe d'objets pour avoir une forme réaliste et facile à utiliser. Ils doivent pouvoir être reconnus au premier coup d'oeil et identifiés comme faisant partie de la scène naturelle. Cela nécessite d'avoir la faculté de modéliser finement chaque détail selon la volonté de l'utilisateur et en accord avec les règles physiques et botaniques élémentaires régissant ces familles d'éléments.

Ainsi, un aspect directement relatif à la modélisation des éléments présents dans une scène naturelle est la génération de variété. En effet, tous les éléments, au sein d'une même famille, doivent être uniques et différer malgré tout les uns des autres. Cette variété de détails rappelle celle de la nature par le fait que chaque élément ne peut être complètement similaire à un autre, ne serait-ce que par ces conditions de croissance différentes. Les couleurs peuvent varier par le même procédé, une exposition plus ou moins longue au soleil, une hygrométrie élevée ou au contraire basse peuvent influencer sur le développement des détails qui nous intéressent, en suivant à nouveau des règles physiques.

Enfin, le vieillissement est un processus prépondérant dans la nature. Il participe activement au réalisme des scènes naturelles en ajoutant encore plus de disparité entre les éléments. De plus, il permet de situer une scène naturelle dans le temps, dans une saison ou bien même dans un contexte particulier. De nombreuses règles régissent le vieillissement dans la nature, des règles physiques et chimiques entrent en jeu pour le vieillissement des structures présentes dans la nature et des règles botaniques régissent une partie du vieillissement des végétaux.

Contributions

Notre approche de cette problématique est de l'aborder en s'inspirant de la nature. Nous proposons d'interpréter l'impact des différentes règles entrant dans le processus de création de ces éléments en les simplifiant, puis nous les appliquons lors de la création des détails. Dans la suite de nos travaux, nous avons décidé d'explorer les différents niveaux d'échelle constituant les détails d'une scène naturelle. Un niveau relativement petit avec les feuilles d'arbres (de l'ordre du centimètre) et un autre niveau plus important avec les altérations et traces sur des terrains (de l'ordre du mètre).

Les premiers éléments traités lors de nos travaux sont les feuilles de végétaux qui sont particulièrement présentes dans les scènes naturelles. Du brin d'herbe à la feuille d'un arbre en passant par les feuilles d'une plante ornementale, ces dé-

tails constituent la majeure partie des végétaux d'une scène naturelle. C'est aussi un élément très changeant, permettant de voir un large spectre de couleurs, de formes et de types de vieillissement. Nous abordons ce problème via un modèle topologique de croissance : le 2-G-Carte L-Système. Ce modèle permet d'appliquer les règles de création d'un L-Système à une 2-G-Carte qui représente une subdivision 2D de l'espace. Grâce à ce modèle, nous pouvons décrire la croissance d'une feuille avec une grammaire dédiée qui définira entièrement les étapes de création basées sur la botanique de base. Nous rajoutons la possibilité d'utiliser des scripts pour obtenir une version plus précise des feuilles souhaitées. Ils permettront de générer, avec une même grammaire, des feuilles différentes appartenant toutes à une même espèce. Une fois la forme de la feuille créée nous générons ses couleurs en fonction de sa topologie. La forme et les couleurs d'une feuille peuvent toutes les deux être affectées par un paramètre d'âge qui détermine le stade de vieillissement de la feuille. Au final, chaque espèce de feuille désirée peut donner lieu à la création d'une infinité d'éléments tous identifiables. De plus, chaque feuille peut être représentée à n'importe quelle étape de sa vie et suivant n'importe quelle quantité avec l'assurance que toutes les feuilles générées seront toutes différentes.

Les feuilles d'arbres représentent des détails relativement petits dans une scène naturelle que leur diversité nous permet d'explorer. Pour la suite de nos travaux, nous avons voulu travailler sur un niveau d'échelle complémentaire aux feuilles. Un autre aspect très important pour le réalisme d'une scène naturelle est le sol. En effet, le feuillage constitue la principale partie des végétaux présents, mais le sol comporte une vaste gamme de déformations qui constituent des éléments d'échelle supérieure aux feuilles.

Qu'on se trouve sur un chemin, une route ou bien même sur un sol rocailleux ou boueux, nous pouvons remarquer quantité de traces et d'empreintes laissées par des animaux, des piétons et même des véhicules. Tous ces éléments altérant le sol peuvent avoir différentes formes d'empreintes, de semelles ou de pneumatiques. Nous avons développé un modèle d'altération du sol reposant sur la simulation d'interactions entre le sol et un objet mobile. Ces altérations se composent d'une trajectoire dessinée par l'utilisateur ainsi que de " patrons " d'empreintes associés. Les traces sont réparties sur l'ensemble de la trajectoire tout en prenant en compte les paramètres de contrôle sur la largeur et le nombre de pas présents. La trajectoire est ensuite décalquée sur le sol puis enfoncée suivant les caractéristiques du sujet réalisant ces empreintes : son poids, sa vitesse et la forme des empreintes.

Ces deux études nous permettent de couvrir un large champ d'échelles. En effet, les feuilles couvrent une grande partie des petits détails alors que les traces sont principalement représentées par des altérations plus conséquentes sur des terrains.

Organisation du mémoire

Le premier chapitre de ce document présente les travaux existants en relation avec nos recherches. La première partie de ce chapitre est consacrée à la génération de feuilles, de leur modélisation à leur rendu. Nous décrivons ensuite les modèles utilisés, à savoir les 2-G-Cartes et les L-Systèmes. Enfin, nous présenterons les travaux existants portant sur la génération et l'altération de terrains.

Le second chapitre présentera notre modèle de génération de détails à petite échelle, à savoir, de grandes quantités de feuilles d'arbres. Nous introduisons notre modèle de 2-G-Cartes L-Système utilisé pour générer ces feuilles grâce à une grammaire unique par espèce. La deuxième partie de ce chapitre montera la génération de texture en accord avec notre modèle ainsi que les aspects de vieillissement appliqués aux feuilles générées.

Le chapitre suivant montre nos recherches sur des éléments d'échelle supérieure. Dans ce cadre, nous avons choisi l'altération de terrains en leur appliquant des trajectoires d'empreintes. Nous décrivons le modèle topologique utilisé ainsi que la génération de trajectoire et son application. Enfin, nous présentons les phénomènes découlant de cette application, c'est-à-dire l'enfoncement des empreintes dans les sols meubles ainsi que le transport et le dépôt de matière le long de la trajectoire. Nous présenterons ensuite les résultats obtenus avec notre méthode d'altération des sols.

Chapitre 1

Etat de l'art.

Sommaire

1.1	Éléments fondamentaux	8
1.1.1	Cartes Généralisées	8
1.1.2	Les systèmes de Lindenmayer (L-Systèmes)	14
1.2	Génération de petits détails : le choix des feuilles	21
1.2.1	Génération de feuilles	21
1.2.2	Rendu de feuilles	34
1.3	A une échelle supérieure : le choix des altérations de terrains	42
1.3.1	Altérations de terrains	42

Chapitre 1

Etat de l'art.

Les différentes échelles de détails sont présentes dans la nature sous une multitude d'aspects. Les scènes naturelles offrent des détails très petits comme des brindilles ou des gravillons ou bien plus importants comme les arbres constituant une forêt.

Nous avons fait le constat que la végétation tient un rôle prépondérant dans ce type de scène, plus particulièrement les feuilles qui constituent les principales parties visibles d'un végétal. Dans notre but d'explorer les détails dans les scènes naturelles à différentes échelles, les feuilles sont un très bon exemple de petits détails se trouvant à une multitude d'endroits dans la nature. De plus, elles proposent une grande variété de formes de couleurs ainsi qu'un vieillissement qui les altère profondément. Ces éléments nous permettent de créer des détails en parcourant tout un spectre de formes et de taille de détails différents. Ainsi, la section 1.2 présente un survol des techniques de génération de feuilles.

Un autre élément visuellement très important dans les scènes naturelles est le sol et toute la collection de déformations qui peut lui être apportée. Afin d'explorer un niveau supérieur d'échelle de détails, nous avons fait le choix d'étudier les empreintes et traces laissées sur un terrain. Ces traces possèdent un grand éventail de formes et de tailles en restant à une échelle assez grande. Nous présentons dans la section 1.3 les différentes approches utilisées pour l'altération des terrains.

Dans un premier temps, nous présentons les différents outils utilisés dans nos méthodes, à savoir, les 2-G-Cartes et les L-Systèmes. Ces modèles sont présentés dans la section section 1.1.

1.1 Eléments fondamentaux

1.1.1 Cartes Généralisées

Les cartes généralisées sont un modèle topologique de représentation par frontières (ou BREP pour *Boundary Representation*). Elles sont une extension des cartes combinatoires (primales et duales) [Tut84] offrant la possibilité de représenter des subdivisions d'objets orientables ou non et pouvant être ouverts. Les cartes généralisées de dimension 2 peuvent être comparées à la structure de données des quad-edge [GS85] qui représente n'importe quelle subdivision de l'espace en 2 dimensions. La démonstration de cette équivalence est présentée dans [Lie91].

1.1.1.1 Description du modèle

Dans nos méthodes, nous utiliserons le modèle des 2-G-Cartes, c'est-à-dire une carte généralisée en dimension 2. Préalablement [Lie88] et [CL92] traitent de l'évolution des surfaces. Ces travaux proposent de faire évoluer une surface en appliquant des opérations topologiques sur un modèle géométrique. Puis le modèle des 2-G-Cartes est présenté dans [Lie89] et [Lie94a].

Nous avons, grâce aux 2-G-Cartes, la possibilité de représenter des subdivisions d'objets en dimension 2 avec ou sans bord et orientable ou non. L'espace est subdivisé en cellules de différentes dimensions, la dimension 0 correspond aux sommets, la dimension 1 aux arêtes et la dimension 2 aux faces.

Définition

Ce modèle possède un unique type d'élément, le brin, représentant une demi-arête. Sur ce brin sont appliquées des involutions visant à assembler les brins en arêtes puis en faces. L'avantage de ce modèle est sa définition homogène pour toutes les dimensions. En effet, seul un type d'élément est utilisé ce qui permet de simplifier les algorithmes ainsi que la définition du modèle.

Une 2-G-Carte est donc définie comme étant un quadruplet $(\beta, \alpha_0, \alpha_1, \alpha_2)$ avec ces différentes propriétés :

- β : un ensemble fini de brins.
- $\forall i \in \{0, 2\}$, α_i est une involution sur β .
- $\forall (i, j)$, $0 \leq i \leq i + 2 \leq j \leq 2$, $\alpha_i \circ \alpha_j$ est une involution.
- $\forall b \in \beta$, $\alpha_0(b) \neq b$ et $\alpha_1(b) \neq b$

Les brins étant l'ensemble des éléments de base de la 2-G-Carte, les opérations

d'assemblage sont les involutions définies sur cet ensemble. Ces involutions sont les suivantes :

- α_0 : involution qui permet de former une arête à partir de deux brins.
- α_1 : involution permettant de coudre entre elles deux arêtes.
- α_2 : involution utilisée pour coudre deux faces le long d'une arête.

De plus les involutions α_0 et α_1 ne doivent pas comporter de points fixes. C'est-à-dire que pour un brin b_1 , $\alpha_0(b_1) \neq b_1$ et $\alpha_1(b_1) \neq b_1$. Intuitivement cette dernière règle exige que les arêtes soient complètes et que les faces soient fermées.

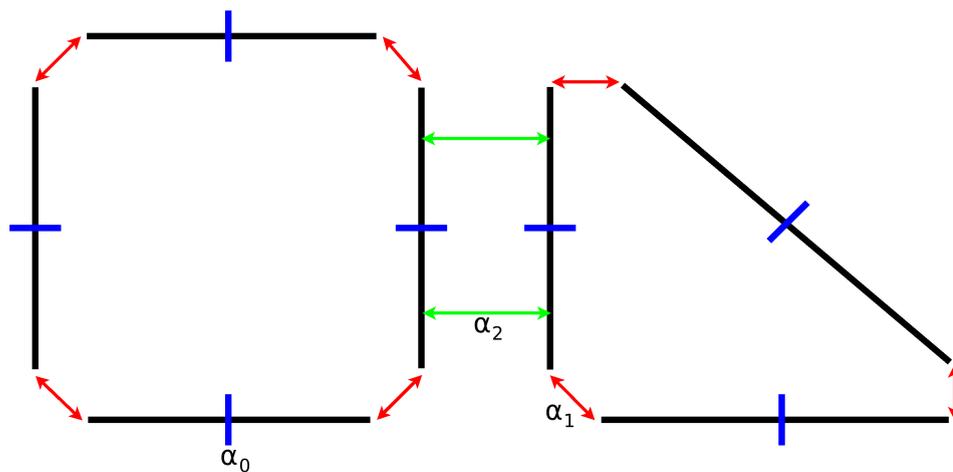


FIGURE 1.1 – La représentation topologique d'une 2-G- Carte avec ses différentes involutions.

Nous pouvons observer sur la figure 1.1 les différentes involutions composant une 2-G- Carte. Nous trouvons en bleu les involutions α_0 liant deux brins en une arête, en rouge les α_1 liant deux arêtes. Enfin, en vert, les α_2 liant les bords de chaque face. Sur cette figure les points fixes par α_2 ne sont pas représentés. En effet chaque arête “ a ” qui ne comporte pas de voisine doit avoir un α_2 tel que $\alpha_2(a) = a$. C'est-à-dire que par l'involution α_2 l'image d'une arête a sera elle-même.

1.1.1.2 Outils des Cartes Généralisées

Le modèle des 2-G- Cartes nous offre des outils puissants pour la modélisation géométrique. Parmi eux, ceux qui nous intéressent principalement sont les processus permettant de parcourir la structure. Avec un brin donné, il est très facile de sélectionner ses éléments voisins en parcourant les éléments appartenant à une même cellule. Nous utilisons principalement deux propriétés du modèle des 2-G- Cartes afin d'accélérer les recherches et les opérations sur nos objets : les bords et les cellules.

Les bords :

Les bords de la 2-G-Cartes sont facilement identifiables. Il suffit de repérer les brins ayant un point fixe par α_2 . En effet, les points fixes sont autorisés pour cette involution au contraire de α_0 et α_1 . Nous avons donc la possibilité de parcourir aussi les composantes connexes de la 2-G-Carte.

Les cellules :

Une cellule est une subdivision de l'espace, les cellules les plus classiques sont les sommets, les arêtes et les faces. Chacune de ces cellules est atteinte en parcourant une orbite qui est une composition d'involutions appliquées à un brin. Nous présentons ici les cellules citées plus haut ainsi que l'orbite qui leur est associée :

- Sommet : $\langle \alpha_1, \alpha_2 \rangle$.
- Arête : $\langle \alpha_0, \alpha_2 \rangle$.
- Face : $\langle \alpha_0, \alpha_1 \rangle$.

Ces différentes cellules sont illustrées dans la Figure 1.2.

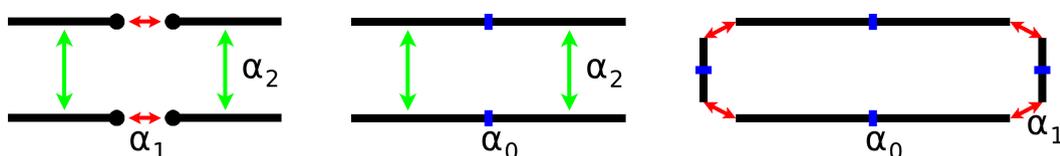


FIGURE 1.2 – Nous pouvons voir sur la gauche la cellule sommet, au centre la cellule arête et sur la droite la cellule face.

L'utilisation de ces orbites permet de grandement accélérer le parcours des éléments d'une 2-G-Carte. Nous pouvons par exemple facilement atteindre les faces connexes à un sommet. Il suffit de parcourir l'orbite sommet et de récupérer les faces liées aux brins atteints.

1.1.1.3 Autres modèles de représentation topologique

Chaînes de cartes :

Les chaînes de cartes [EL92, Lie94b] sont une extension des cartes généralisées et permettent de modéliser des objets non-variétés comme des complexes cellulaires. Les chaîne de dimension n , aussi appelée n -chaîne, possède une définition basée sur les n -G-Cartes et une structure équivalente à la notion de complexe cellulaires.

Une n -chaîne sera composée de cellules de dimension p ($p \in [0, n]$) représentées par une p -G-Cartes. Les brins d'une même p -cellule seront liés par les involutions classiques α des cartes généralisées qui seront appelées les liaisons intracellulaires.

Les relations de bords entre ces différentes cellules devront par contre être traitées par d'autres applications qui seront appelées des liaisons intercellulaires σ . De ce fait, chaque p -cellule sera associée aux q -cellules de dimensions inférieures ($0 \leq q \leq p \leq n$) par des relations spécifiques σ . Pour une p -cellule, la relation σ_q^p donnera l'image d'un brin lui appartenant comme étant soit un brin de la q -G-Carte soit une valeur indéfinie si le bord de la p -cellule est incomplet.

Ce modèle à l'avantage d'être très complet en offrant la possibilité de composer des objets avec des G-Cartes de différentes dimensions. Nous pouvons voir cependant que sa complexité s'en trouve fortement augmentée :

```
class Brin
{
    //dimension de la cellule à laquelle appartient ce brin
    int p;

    //Liaisons intracellulaires
    Brin* alpha[0..p-1];

    //Liaisons intercellulaires
    Brin* sigma[0..p-1];

    //Inverses des liaisons intercellulaires
    Brin* sigma_inv_prem[p+1..n];
    Brin* sigma_inv_suiv[0..p-1];

    //Autres informations (plongement, marquage ...)
}
```

Cette implémentation de base possède donc $2p + n$ pointeurs. Nous pouvons la comparer à l'implémentation de base des n -G-Cartes qui ne possède que n pointeurs :

```
class Brin
{
    //Liaisons intracellulaires
    Brin* alpha[0..n];

    //Autres informations (plongement, marquage ...)
```

} _____

La gestion des chaînes de carte ajoute donc $2p$ pointeurs. Plus précisément, dans le cas de la comparaison des 2-Chaînes et des 2-G-Cartes, les 2-Chaînes possèdent pour un brin appartenant à l'orbite 0 (sommet) 2 pointeurs. Pour l'orbite 1 (arêtes) 2 pointeurs et pour l'orbite 2 (faces) 6 pointeurs. Dans ce même cas, la 2-G-Carte possède pour un brin donné seulement 2 pointeurs. Pour une structure plus conséquente en espace de stockage, nous avons de plus les mêmes performances sur les algorithmes appliqués lors des parcours et recherches.

Les objets que nous modéliserons, à savoir des feuilles d'arbres, ne possèdent qu'une très faible épaisseur. Les chaînes de cartes auraient pu nous donner accès à la modélisation de nervures volumiques en les modélisant sous forme de 3-G-Cartes ou bien de nervures sans largeur en les modélisant sous la forme d'une 1-G-Carte. Pour représenter des feuilles considérées comme "plates" Nous ne choisirons donc pas de modèle gérant les subdivisions de l'espace en plus de deux dimensions pour les représenter. De plus, pour simuler les nervures de différentes largeurs nous avons besoin que ces nervures soient elles aussi des subdivisions en deux dimensions de l'espace. Enfin, sans nervure possédant une largeur, nous ne pouvons pas définir de collage de faces sur un des côtés de cette face. Ce processus aurait pu être simulé avec des arêtes décrites par une 1-G-Carte, mais aurait apporté beaucoup de complication aux règles de la grammaire.

Pour ces raisons, nous avons choisi le modèle des 2-G-Cartes. En effet, ce modèle correspond parfaitement aux attentes que nous avons dégagées concernant la modélisation des feuilles et nous permet une facilité d'utilisation dans les règles que nous mettrons en place dans la grammaire.

Cartes multirésolutions :

Les cartes multirésolutions introduites dans [KCB07, KCB09] permettent de représenter des objets géométriques avec plusieurs niveaux de détails. A chaque ajout d'un niveau de détail, des brins et des relations sont ajoutés à la 2-Carte multirésolution. Ainsi, pour chaque niveau de subdivision, l'objet sera représenté par une carte combinatoire. Ces représentations nous permettent d'utiliser les outils classiques des cartes combinatoires (Parcours efficaces, structure légère et homogénéité du modèle). Ce modèle permet de garder, pour chaque niveau de subdivision, un maillage utilisable avec l'assurance qu'il n'y a pas de trou. Il peut, de plus, appliquer des déformations sur n'importe quel niveau de subdivision voulu.

Dans le cadre de la génération de traces et d'empreintes sur des terrains, nous procéderons au raffinement du terrain pour pouvoir y inclure les déformations dues aux traces laissées par les objets simulés (pieds, véhicules, animaux). Cependant, cet ajout de résolution est effectué uniquement aux endroits où des traces vont apparaître et nous n'avons aucun besoin d'accéder aux niveaux de résolutions supérieurs.

En effet, les niveaux de détails que nous mettrons en place seront uniquement régis par la prise de vue et donc, sans gestion de l'animation, nous ne devons pas modifier le raffinement ou en utiliser les différentes échelles. En effet, dans notre cas le raffinement prend comme paramètre la taille d'une arête projetée sur l'écran. Si cette taille passe en dessous d'un seuil défini, alors cette arête ne sera plus subdivisée. Dans ce sens, les niveaux de détails sont gérés par notre modèle uniquement en fonction de la prise de vue.

Le modèle des cartes multirésolutions s'avère très efficace pour gérer des déformations en temps réel et des niveaux de détails dans des animations. Nous utiliserons dans notre cas une 2-G-Carte avec un mécanisme de subdivision pour à la fois respecter la contrainte de résolution maximale et la nécessité d'épouser au mieux la forme d'une empreinte. Pour un tel traitement, les 2-G-Cartes sont moins gourmandes en espace de stockage du fait qu'il y a moins de brins à stocker.

1.1.1.4 Conclusion

Cette structure de données nous permet de travailler sur des subdivisions de l'espace en 2 dimensions qui sont parfaitement adaptées aux objets que nous aurons à manipuler, c'est-à-dire les objets orientables avec bords. Les cartes généralisées de dimension 2 nous offrent en outre une grande facilité et simplicité d'implémentation.

De plus, ce modèle est homogène pour n'importe quelle dimension grâce à l'utilisation d'un unique élément, le brin. Ce modèle nous apporte aussi un accès direct au voisinage de n'importe quelle cellule (faces adjacentes, sommets ou bien arêtes).

Enfin, ce modèle présente une gestion simple et efficace de la géométrie en séparant de façon claire la topologie portée par les brins et le plongement géométrique porté par les sommets. Ce modèle requiert toutefois une place mémoire plus grande qu'un maillage classique du fait de la redondance des informations.

Afin de pouvoir gérer la croissance d'un tel modèle topologique, nous avons étudié les systèmes de Lindenmayer. Ces L-Systèmes nous permettent, grâce à une grammaire formelle, d'appliquer des règles de croissance au modèle géométrique désiré.

1.1.2 Les systèmes de Lindenmayer (L-Systèmes)

Les systèmes de Lindenmayer ou L-Systèmes sont introduits dans [PL96]. Ils sont composés de grammaires formelles permettant de modéliser le processus de développement de plantes. Les règles de la grammaire sont appliquées en parallèle à partir d'un élément de base appelé axiome.

Un L-Système se base sur une grammaire formelle qui permet de modéliser

entièrement le développement et la croissance d'un système arborescent. Les différentes règles de cette grammaire sont appliquées en parallèle les unes des autres contrairement aux grammaires de Chomsky où elles sont appliquées de façon séquentielle. Le point de départ de cette génération est un axiome permettant de connaître où et comment les règles s'appliqueront.

Il existe un grand nombre de types de L-Systèmes. Cependant, ils sont tous dérivés d'un L-Système de base nommé D0 L-Système.

1.1.2.1 Le D0 L-Système

Le D0 L-Système est le L-Système de base développé par Aristid LINDENMAYER dans [PL96]. C'est un système déterministe : pour un mot, une seule règle peut être appliquée. Il est, de plus, indépendant du contexte.

Un D0 L-Système peut être décrit de la façon suivante :

- Un alphabet V : l'ensemble des variables du L-Système (on note V^* l'ensemble des mots formés à partir de V et V^+ une restriction de V^* aux mots non vides),
- Un axiome ω ($\omega \in V$) : c'est l'état initial du L-Système,
- Un ensemble de règles de production $P \subset V \times V^*$.

Le D0L-système est alors défini par le triplet : (V, ω, P) .

Voici un exemple simple de D0L-système :

- Alphabet : a,b
- Axiome : $\omega = a$
- R1 : $a \rightarrow b$
- R2 : $b \rightarrow ab$

Au fil des itérations, voici les mots formés par ce D0L-système : $a, b, ab, bab, abbab, bababbab, abbabbababbab, etc...$

1.1.2.2 Interprétation graphique des L-Systèmes

Avant de détailler les autres types de L-Systèmes, attardons-nous sur la possibilité de traduire en représentation graphique une telle génération.

Pour pouvoir afficher graphiquement un D0L-système sous forme de segments de droite, plutôt que d'afficher une suite de lettres appartenant à V , le principe de la tortue LOGO est utilisé. Cette tortue, introduite dans le langage LOGO par

Seymour PAPERT, a pour but de faire comprendre la programmation graphique aux enfants (traçage de lignes, segments, arcs de cercle, etc.). Il s'agit d'une sorte d'automate se déplaçant dans le plan et ayant un système d'axe propre (H,L,U) représentant respectivement la direction de la tortue (Head), la direction de la gauche (Left) et la direction du haut (Up). Cette tortue peut se voir affecter des ordres simples comme "tourner à droite de 90°" ou bien "avancer de 3 unités".

La tortue LOGO possède son propre état sous forme d'un triplet (x,y,α) , avec (x,y) sa position actuelle et α l'angle décrivant la direction de sa tête. Pour être intégrée dans un L-Système, la tortue a été pourvue d'un formalisme pour décrire ses différents mouvements dans la grammaire :

- $F(s)$: avancer de s unités dans la direction de la tortue et tracer le segment,
- $f(s)$: avancer de s unités sans traçage,
- $+(\theta)$ tourner à gauche d'un angle θ autour de U,
- $-(\theta)$ tourner à droite d'un angle θ autour de U,
- $\wedge(\theta)$ tourner à gauche d'un angle θ autour de L,
- $\&(\theta)$ tourner à droite d'un angle θ autour de L,
- $/(\theta)$ tourner à gauche d'un angle θ autour de H,
- $\backslash(\theta)$ tourner à droite d'un angle θ autour de H.

De plus, deux opérateurs sont introduits pour permettre de représenter les structures arborescentes :

- [Empile l'état de la tortue,
-] Dépile l'état depuis la pile d'états et l'affecte à l'état de la tortue.

Voici un exemple de grammaire traitée avec un DOL-système : la courbe de VON KOCH (ou flocon de neige de VON KOCH). Cette courbe repose sur la grammaire suivante :

- ω : $F-(120)F-(120)F$
- $p1$: $F \rightarrow F+(60)F-(120)F+(60)F$

Pour chaque itération, chaque segment F du flocon est transformé en quatre autres segments. Le résultat de la génération de cette grammaire est représenté sur la Figure 1.3.

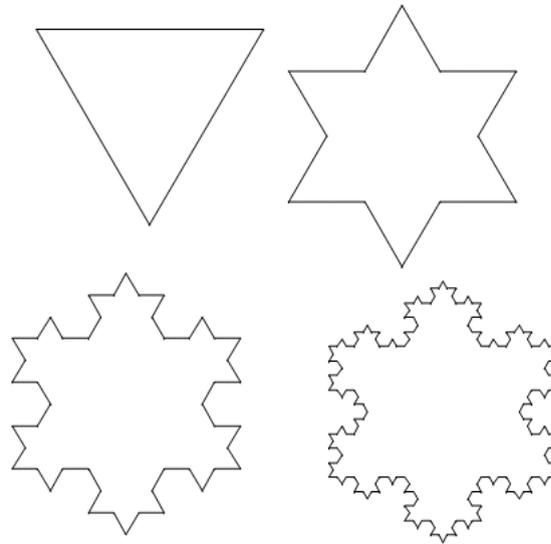


FIGURE 1.3 – Flocon de neige de VON KOCH. De gauche à droite et de haut en bas nous pouvons observer le résultat pour 0, 1, 2 et 3 itérations

1.1.2.3 Le L-Système paramétrique

Les L-Systèmes paramétriques sont présentés par Prusinkiewicz *et al.* dans [Pru04] et sont réutilisés par Rodkaew *et al.* dans [RLFS02]. Ils ont été conçus pour pouvoir mieux contrôler l'application des règles de production. Ces L-Systèmes permettent de faire varier certaines composantes de la grammaire en introduisant des paramètres. Ceux-ci peuvent être de types différents, tels que la longueur de certains segments, leur épaisseur ou encore un angle α .

Voici un exemple de L-Système paramétrique dont le résultat apparaît sur la figure 1.4 :

- $\omega : !(3)F(1,1)$
- $p1 : F(s, t) : t=1 \rightarrow F(s,2)[-!(1)F(s,1)][+!(1)F(s,1)]F(s,2)!(1)F(s,1)$
- $p2 : F(s, t) : t=2 \rightarrow F(2*s,2)$
- $p3 : !(w) : w < 2 : !(3)$

Nous pouvons remarquer que chaque règle possède deux paramètres “s” et “t”. “s” représente la longueur du segment et “t” indique le statut de noeud ou un d’internoeud d’un segment ($t=1$ ou $t=2$ respectivement). De plus, ici, un nouvel opérateur “!(e)” est introduit, son rôle est de fixer l’épaisseur de la ligne à “e”. Si le segment F est un noeud, alors c’est la règle p1 qui est appliquée ($t=1$) sinon c’est la règle p2 qui sera prise en compte ($t=2$).

La règle p3 quant à elle porte sur l’épaisseur d’un trait. Elle est appliquée à chaque itération sur tous les segments composant le L-Système si son épaisseur est inférieure à deux, alors son épaisseur est incrémentée. Toutes sortes de paramètres

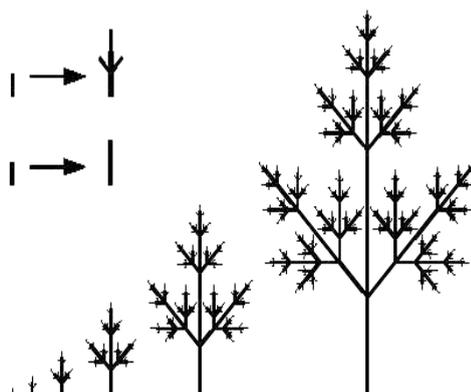


FIGURE 1.4 – L-Système paramétrique.

peuvent être définis de cette façon, comme l'angle des internoeuds qui augmente à chaque itération prenant en compte le poids des branches rajoutées.

1.1.2.4 L-Systèmes dépendants du contexte

Dans les différents L-Systèmes vus précédemment, aucun ne tient compte du contexte pour appliquer les règles. Or il peut être très intéressant d'appliquer une règle uniquement si certaines conditions sont réunies. Ces conditions peuvent être de tous types, par exemple pour modéliser le fait que les branches ne peuvent pas pousser de chaque côté d'un noeud (Fig. 1.5).

Nous pouvons distinguer deux types de L-Systèmes sensibles au contexte :

- 2L-Système : les règles peuvent s'appliquer dépendant du contexte gauche ou droit de l'ensemble des prédécesseurs,
- 1L-Système : les règles peuvent s'appliquer dépendant soit du contexte gauche, soit du contexte droit de l'ensemble des prédécesseurs.

Pour ce type de L-Système, les règles utilisées sont de la forme :

$$A_g < A > A_d \rightarrow M$$

Le mot A sera changé en M, si et seulement si le mot A est entouré de A_g et A_d , avec A_g et A_d étant respectivement le contexte gauche et le contexte droit de A.

Par exemple, le contexte droit permet de simuler une information se propageant de la pointe vers la racine (signal basipète) et le contexte gauche une information qui se propage de la racine vers la pointe (signal acropète) (Fig. 1.5).

Sur la figure 1.5 nous pouvons voir le résultat du L-Système suivant :

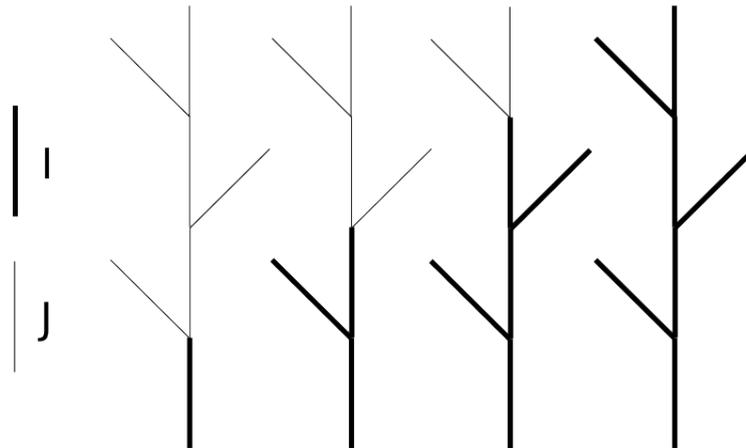


FIGURE 1.5 – Propagation depuis la racine des segments “I” dans un L-Système dépendant du contexte.

- $\omega : I[J]J[J]J[J]J$
- $p1 : I < J \rightarrow I$

Par propagation depuis la racine, les brins J se transforment en brin I.

1.1.2.5 L-Système stochastique.

Un aspect de la génération par L-Système reste très peu naturel. Il s’agit de la régularité de sa génération. Pour palier ce problème, [RLFS02] utilise un L-Système stochastique permettant d’introduire un aléa dans l’utilisation des règles. Bien entendu, ce type de L-Système n’est pas déterministe, car il est possible de déterminer plusieurs transformations pour un symbole donné.

Pour arriver à ce résultat, il faut ajouter à la définition du L-Système une fonction de distribution (notée p). C’est maintenant un quadruplet, et non un triplet qui définit le L-Système. Il est de la forme : (V, ω, P, p) .

p est la fonction de distribution qui fait la correspondance entre l’ensemble des règles de production et l’ensemble des probabilités que ces règles soient appliquées.

Exemple de L-Système stochastique :

- $\omega : F$
- $p1 : F \rightarrow F[+F]F[-F]F : 1$
- $p2 : F \rightarrow F[+F]F : 1$
- $p3 : F \rightarrow F[-F]F : 5$

Les chiffres en fin de règle désignent les probabilités que cette règle se produise. En renouvelant le processus, on trouvera alors des résultats différents (fig. 1.6).



FIGURE 1.6 – Trois générations avec le même L-Système stochastique.

1.1.2.6 Map L-Système.

Les Map L-Systèmes proposés par [PL96] peuvent décrire des systèmes possédant des cycles. Il est alors naturel de construire des surfaces représentées par leurs bords tout aussi facilement qu'avec un L-Système classique. Ces surfaces peuvent être de différents types et décrire des structures cellulaires.

Exemple d'un Map L-Système simple (résultat fig. 1.7) :

- ω : ABAB
- p1 : $A \rightarrow B [-A] [+A] B$
- p2 : $B \rightarrow A$

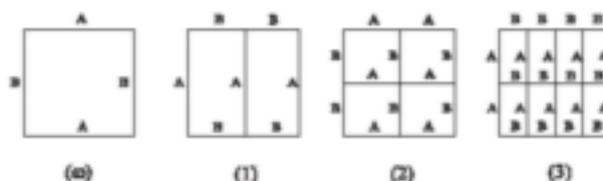


FIGURE 1.7 – Map L-Système simple.

Nous pouvons constater que la définition d'un Map L-Système est le même qu'un L-Système classique. Seule l'interprétation graphique est différente à cause des cycles définissant les morceaux de surface, une telle surface est appelée carte.

Un Map L-Système possède les caractéristiques suivantes :

- Une map est un ensemble de régions.
- Chaque région est entourée d'une frontière constituée d'un ensemble fini d'arêtes reliées par leurs sommets.
- Chaque arête possède un ou deux sommets. Si elle a un seul sommet alors elle forme une boucle.
- Les arêtes ne peuvent pas se croiser sans former un nouveau sommet et il n'y a pas de sommet libre indépendant d'une arête.
- Chaque arête fait partie de la frontière d'une région.
- L'ensemble des arêtes doit être connecté, c'est-à-dire qu'il n'y a pas de trou

dans la carte.

Le Map L-Système peut, grâce à ces caractéristiques, modéliser la croissance de subdivisions 2D de l'espace. Cependant, son formalisme rend les grammaires mises en jeu relativement complexes.

1.1.2.7 Conclusion

Les différents types de L-Systèmes présentés dans cette partie permettent de définir la croissance de systèmes linéaire avec beaucoup de précision. En effet, l'application des règles de production permet de faire croître une structure arborescente de sa base jusqu'à ses ramifications les plus éloignées. Les nombreuses améliorations ajoutées aux L-Systèmes basiques comme les paramètres, la dépendance au contexte et les aspects aléatoires permettent une définition encore plus précise du phénomène de croissance, et donc, du modèle de végétaux désiré.

Malgré cette précision dans la définition, les L-Systèmes sont peu adaptés pour décrire la croissance de structures en dimension 2. Effectivement, même les Map L-Systèmes introduits pour pallier ce manque n'offrent pas de résultats vraiment satisfaisants. Les règles sont difficilement applicables de façon cohérente du fait des mécanismes automatiques de coutures des arêtes de ce modèle.

1.2 Génération de petits détails : le choix des feuilles

1.2.1 Génération de feuilles

Dans le cadre de la synthèse d'images réalistes, la modélisation de végétaux est un sujet complexe. En effet, la grande diversité des végétaux et de leurs feuillages fait que la création d'outils capables de proposer leur modélisation générique et exhaustive est un véritable défi. Les personnes travaillant sur ce sujet se restreignent le plus souvent à un domaine spécifique comme les L-Systèmes, les systèmes de particules ou bien la simulation de processus chimiques.

Nous verrons dans une première partie les différentes méthodes développées pour la modélisation de feuilles, une majorité d'entre elles reposent sur les L-Systèmes. Ensuite, nous verrons les méthodes proposées pour effectuer le rendu de feuilles.

1.2.1.1 Modélisation à partir de L-Systèmes

Les scènes naturelles sont une des catégories d'image réalistes les plus complexes à créer en raison de la nature botanique et physique des plantes. En effet,

une méthode pouvant modéliser tous les types de plantes est difficile à mettre en oeuvre et requière d'intégrer de nombreux facteurs.

Dans cette optique, les méthodes de modélisation à partir de L-Systèmes permettent de créer une structure botaniquement réaliste. En effet, Les L-Systèmes représentent parfaitement la croissance d'une plante et permettent de représenter finement le réseau de nervures d'une feuille. Ce réseau sera évidemment composé de segments de droite si les L-Systèmes sont utilisés seuls. C'est pourquoi cette méthode est en générale couplée avec des modèles classiques utilisés en modélisation géométrique.

L-Systèmes et Cylindres généralisés.

La combinaison de L-Systèmes et de cylindres généralisés proposée par [PMKL01] permet de créer des feuilles et des tiges de plantes.

Cette méthode utilise les cylindres généralisés (Fig. 1.8), c'est-à-dire l'extrusion suivant un chemin quelconque (une courbe) d'une section quelconque (courbe fermée, cercle, ellipse, etc..) pouvant varier (en forme et en taille) le long du chemin.



FIGURE 1.8 – Un exemple de cylindre généralisé.

Une fois mis en place, le cylindre doit être échantillonné pour pouvoir être affiché. La méthode classique est de prendre un pas d'échantillonnage et de transformer le cylindre en un ensemble de sections le long du chemin. Le pas doit être choisi assez petit pour ne pas faire apparaître à l'affichage la géométrie du cylindre discrétisé. De plus, un autre pas d'échantillonnage est choisi pour la section, il est choisi comme pour le chemin, assez petit pour rendre la modélisation fine et sans angle.

Cette méthode étant basée sur des cylindres généralisés, il lui est difficile de modéliser des feuilles de façon précise. Les cylindres généralisés permettent de définir assez finement le réseau de nervures d'une feuille, mais n'autorisent pas la création du limbe de celle-ci. Cette méthode est plus appropriée à la création de tiges et de branches qu'à celle de feuilles.

L-Systèmes et algorithmes génétiques.

Pour générer des feuilles [RLFS02] utilise un L-Système à base de fonctions tag et un algorithme génétique pour faire croître le L-Système. Les fonctions tags remplacent les paramètres que l'on aurait pu mettre dans un L-Système paramétrique classique. Elles peuvent être n'importe quelle fonction usuelle. La conversion de ces fonctions tag en paramètres est ensuite effectuée (Fig. 1.9).

Ce sont ces paramètres qui seront pris en compte par la tortue logo lors de la génération et l'affichage du L-Système.

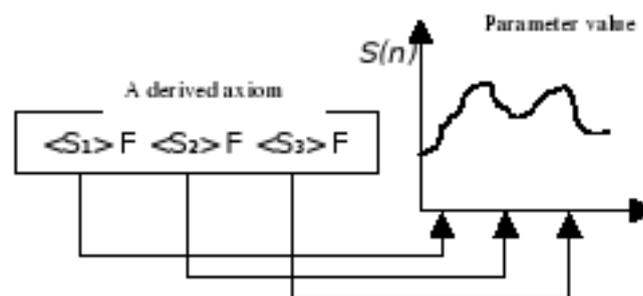


FIGURE 1.9 – Conversion des fonctions tag en paramètres.

Les résultats obtenus sont visibles sur la figure 1.10. Nous voyons à gauche le squelette généré par les règles puis les trois squelettes de droite sont des déformations du premier par ajustement des fonctions tag.



FIGURE 1.10 – Divers squelettes générés avec les fonctions tag.

[RLFS02] utilise ensuite un algorithme génétique pour faire que la croissance du L-Système épouse le plus possible la forme de la feuille que l'on souhaite modéliser. Pour ce faire, l'utilisateur doit fournir une image donnant le contour de la feuille.

L'algorithme génétique utilisé est basé sur l'algorithme proposé par John Holland de l'université du Michigan. Il est basé sur la correspondance entre un individu et le contour souhaité. À chaque étape de la génération, plusieurs individus sont générés et seul celui qui remplit au mieux les critères de remplissage du contour de la feuille est utilisé pour produire la génération suivante. Les fonctions tag étant utilisées comme critère de sélection, leurs valeurs s'affinent au fur et à mesure des étapes pour donner un résultat le plus proche possible du contour de la feuille réelle.

Les résultats de cette méthode sont très proches du contour d'une feuille réelle, cependant le réseau de nervure manque de réalisme (Fig. 1.11). Ces nervures

s'étendent quoiqu'il arrive jusqu'au contour de la feuille fourni par l'utilisateur. Ce contour étant unique, chaque feuille sera similaire pour chaque génération. De plus, les nervures générées par cette méthode sont peu réalistes dans le sens où elles sont beaucoup trop grosses et ne comportent aucune ramification. En effet, [RLFS02] ne peut pas générer de squelettes très compliqués autres qu'une nervure principale et des nervures secondaires.



FIGURE 1.11 – Rendu des résultats de la méthode de [RLFS02]

1.2.1.2 Modélisation basée image.

Pour la méthode de reconstruction de plantes basée images, [QTZ⁺06] propose une technique semi-automatique. L'utilisateur doit fournir un jeu de photographies chacune prise avec une légère rotation par rapport à la précédente, elles fourniront le nuage de points 3D représentant la plante photographiée.

La figure 1.12 décrit la méthode utilisée par [QTZ⁺06].

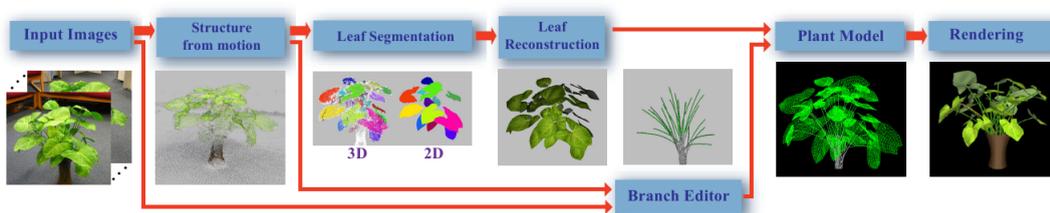


FIGURE 1.12 – Pipeline de la méthode utilisée dans [QTZ⁺06] pour générer un végétal.

Une fois le nuage de points obtenu, il est segmenté à l'aide des photographies pour obtenir uniquement les points correspondants aux feuilles. Ici, l'utilisateur peut agir sur la séparation des points à partir du nuage 3D ou des images 2D fournies. Un modèle de feuilles génériques déformables est ensuite appliqué pour obtenir chaque feuille séparément. La même méthode est ensuite utilisée pour fabriquer les tiges de la plante. Les résultats de cette méthode possèdent une complexité et une allure visuelle proche du modèle pris en photo (Fig. 1.13). De plus, la nature géométrique du modèle obtenu permet de facilement déformer la plante.

FIGURE 1.13 – résultat de la modélisation basée image de [QTZ⁺06].

Plus spécifiquement, la méthode employée pour récupérer les points correspondants aux feuilles est une optimisation de graphe interactive utilisant les informations 3D du nuage de points et les informations 2D des images fournies. De plus pour ne pas surcharger l'utilisateur, [QTZ⁺06] n'utilise qu'une seule extraction de feuille pour générer tout le feuillage (en effet, l'extraction de toutes les feuilles d'une plante prendrait un temps considérable).

La segmentation des feuilles se fait en deux temps : la segmentation grâce à un algorithme de graphes et la retouche de la segmentation par l'utilisateur.

Première phase : construction du graphe.

Chaque point du nuage de points définit un noeud, ensuite les arêtes du graphe sont définies en liant un noeud à ses K plus proches voisins (ici $K=3$). Les arêtes du graphe sont ensuite pondérées par un réel qui définit la probabilité d'appartenance des deux noeuds liés par cette arête à la même feuille. Pour calculer ce coefficient [QTZ⁺06] définit une distance entre les noeuds tenant compte de la distance 2D et de la distance 3D.

La distance combinée entre deux noeuds p et q devient :

$$d(p, q) = (1 - \alpha) \frac{d_{3D}(p, q)}{\sqrt{2}\sigma_{3D}} + \alpha \frac{d_{2D}(p, q)}{\sqrt{2}\sigma_{2D}}$$

Avec :

$d_{3D}(p, q)$, la distance euclidienne 3D entre le point p et le point q .

σ_{3D} , la variance de cette distance.

α , un scalaire de poids, défini à 0,5 par défaut.

$d_{2D}(p, q)$, la distance 2D prenant en compte toutes les images utilisées pour la segmentation.

σ_{2D} , la variance du gradient de couleur sur chaque image.

Le poids de chaque arête est ensuite calculé avec la formule suivante :

$$w(p, q) = e^{-d^2(p,q)}$$

Et le graphe est initialisé comme ceci :

$$w(p, q) = \begin{cases} e^{-d^2(p,q)} & \text{si } d_{3D} < K\sigma_{3D} \text{ et } d_{2D} < K\sigma_{2D} \\ 0 & \text{sinon} \end{cases}$$

Les groupes de points ressortant de cette modélisation apparaissent très différents comme on peut le voir sur la figure 1.14.a . Cependant, il faut l'intervention de l'utilisateur pour affiner la segmentation.

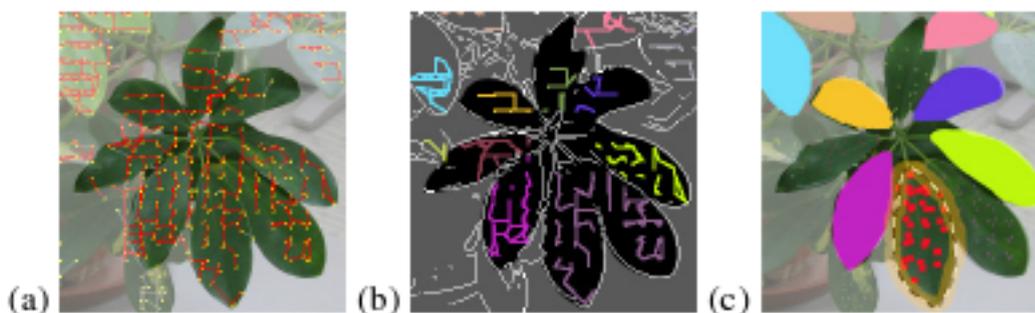


FIGURE 1.14 – Les différentes phases de la segmentation des feuilles.

Deuxième phase : affinage par l'utilisateur.

Une fois les zones définies par le graphe (Fig. 1.14.b) l'utilisateur possède deux actions pour raffiner la segmentation. Il peut choisir un groupe dont la segmentation est acceptable ou bien dessiner une frontière pour raffiner un groupe correspondant à une feuille (Fig 1.14.c). La feuille ainsi obtenue sera utilisée pour la génération du feuillage.

La modélisation basée image des feuilles présentées par [QTZ⁺06] nécessite des traitements complexes et coûteux pour reproduire une feuille à partir d'un modèle réel. De plus, toutes les feuilles des végétaux créés par cette méthode seront complètement identiques et ne reproduiront pas l'hétérogénéité d'un feuillage trouvé dans la nature. Cependant, en utilisant un grand nombre de données directement extraites des photographies de plantes, les résultats de cette méthode sont visuellement convaincants pour l'aspect d'une feuille.

1.2.1.3 Algorithme biologique.

Pour représenter les nervures des feuilles, [RFL⁺05] propose un algorithme prenant en compte la diffusion d'hormones. Cette méthode se base sur le développement des nervures en fonction des sources d'hormones contenues dans la feuille.

Les nervures vont se développer pour se rendre vers les sources d'hormones dans leur proximité. Cette migration a pour conséquence la croissance de la feuille.

Tout d'abord, l'utilisateur doit fournir la forme de base de la feuille (Fig. 1.15). Elle est composée d'un contour (courbe paramétrée) et d'un noeud représentant le départ des nervures (mis sur le pétiole). Dans le cas de feuilles comportant plusieurs lobes, l'utilisateur peut mettre plusieurs noeuds de nervure à la base de la feuille. Les sources d'hormones sont quant à elles placées au hasard sur la feuille grâce à l'algorithme du lancer de fléchettes.

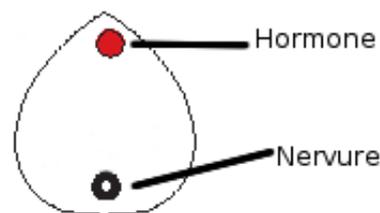


FIGURE 1.15 – Etape initiale de l'algorithme.

Ensuite la croissance de la feuille se fait pendant les itérations de l'algorithme. La figure 1.16 présente un exemple d'itérations de la méthode de [RFL⁺05].

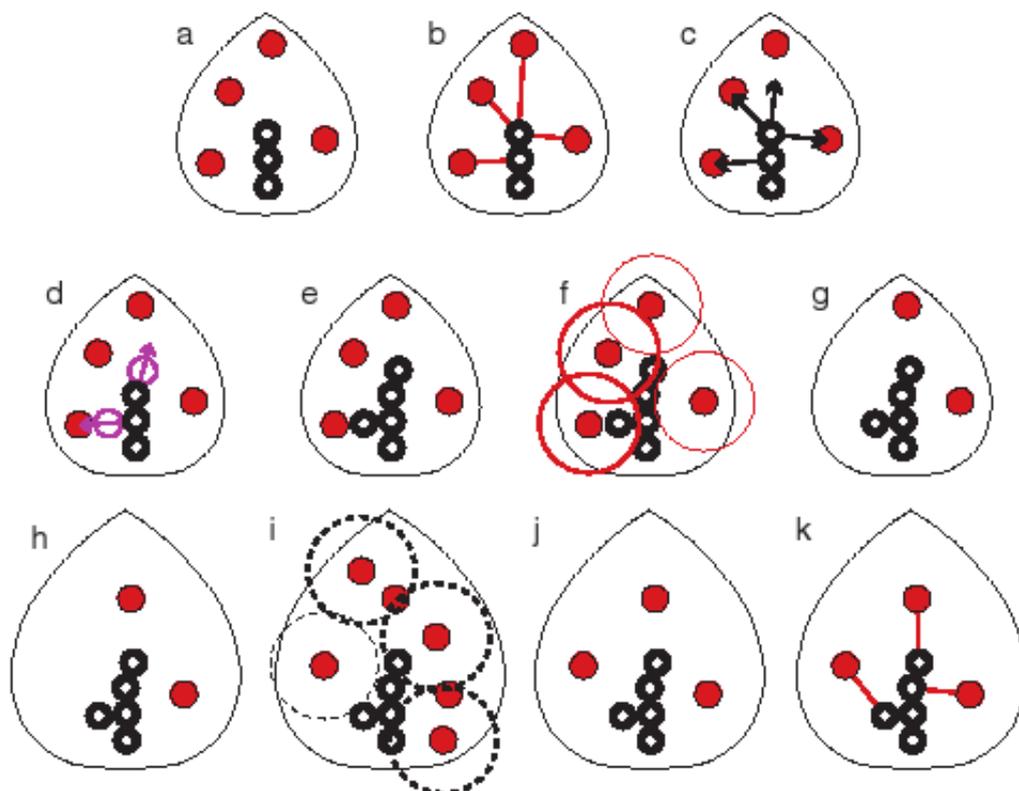


FIGURE 1.16 – Exemple de fonctionnement de la méthode de [RFL⁺05].

Dans la figure 1.16 les disques représentent les sources d'hormones et les cercles représentent les noeuds de nervures. La figure 1.16.a montre l'état courant de

la croissance de la feuille. Ensuite (Fig 1.16.b) chaque source d'hormones se voit associer le noeud le plus proche, des vecteurs normaux sont alors obtenus (Fig. 1.16.c), en faisant une moyenne de ceux-ci on obtient les vecteurs qui permettront de rajouter un noeud de nervures (Fig. 1.16.d). Les nouveaux noeuds sont ensuite ajoutés à la feuille (Fig. 1.16.e) puis les sources d'hormones étant dans un voisinage trop proche des noeuds sont supprimées (Fig. 1.16.f). Les figures 1.16.g et 1.16.h montrent l'accroissement du contour, puis 1.16.i la répartition de nouvelles sources d'hormones et leur suppression si elles sont trop proches des nervures. Finalement 1.16.j présente une nouvelle étape initiale de l'algorithme, qui recommence à la figure 1.16.k.

L'épaisseur des nervures est donnée par la loi de Murray : $Enm = \sigma Enf$ avec Enm l'épaisseur de la nervure mère et Enf l'épaisseur des nervures filles. Pour ce faire, [RFL⁺05] commence par calculer l'épaisseur des nervures les plus jeunes, car elles n'auront pas de nervures filles, ensuite le réseau de nervures est remonté pour calculer récursivement l'épaisseur des nervures mères.

Cette méthode permet de reproduire les nervures des feuilles avec beaucoup de précisions et de réalisme (fig. 1.17). De plus, ce réalisme est d'origine biologique. Cependant, cette méthode offre peu de contrôles à l'utilisateur pour modifier la feuille créée. Elle nécessite de plus de fournir le contour de la feuille sous forme de texture afin de délimiter le cadre de croissance des nervures.



FIGURE 1.17 – A droite nous pouvons voir la photo d'une feuille et à gauche la même feuille générée par [RFL⁺05].

1.2.1.4 Modélisation par squelette.

La modélisation de feuilles lobées fait l'objet d'une méthode développée par [MMPP03]. Dans cette méthode, l'utilisateur doit fournir une silhouette 2D de la feuille à modéliser. Ensuite un squelette est calculé à partir de cette silhouette en suivant l'axe médian, il est représenté avec le modèle des sticky splines de [vOV96]. Ce modèle particulier permet d'avoir des splines avec des relations topologiques. Une fois la silhouette et le squelette terminés (fig. 1.18), la feuille est alors construite.



FIGURE 1.18 – Silhouette et squelette de la feuille lobée.

La création du maillage se fait par balayage du contour et du squelette. La méthode consiste à prendre un point de départ sur le squelette et sur le contour (points s et k Fig. 1.19.a), ensuite le squelette est balayé en parallèle du contour (dans le sens direct) jusqu'à ce que l'on trouve un point terminal sur le squelette (point k_T Fig. 1.19.b). Ce point est lié au contour en prenant l'intersection entre la tangente au point k_T et le contour. Cela nous donne le point s_T (Fig. 1.19.b).

Après avoir défini le point d'encrage s_T le squelette est encore une fois balayé depuis le point k , et les points d'encrages intermédiaires sont calculés en prenant le point du contour le plus proche des points k_j (Fig. 1.19.c et 1.19.d). Ce processus est réalisé sur tout le balayage du squelette pour donner tous les points d'ancrage (Fig. 1.19.e).

Enfin, pour finir de créer le maillage, le squelette est relié au contour en utilisant la paramétrisation de la longueur d'arc du squelette et du contour (Fig. 1.19.f).

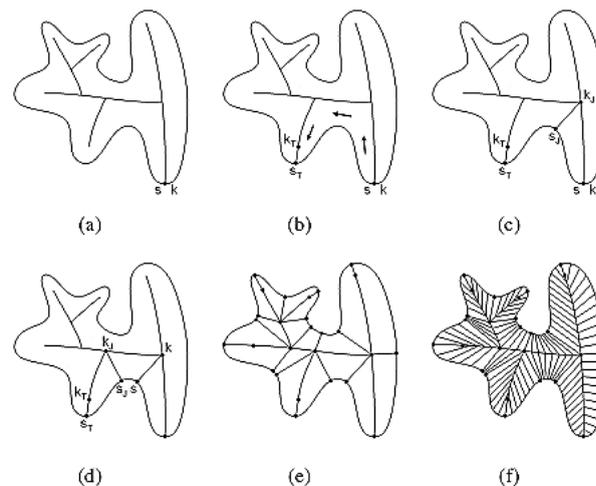


FIGURE 1.19 – Les différentes étapes de la création du maillage.

Cette méthode donne des résultats réalistes (Fig. 1.20), cependant le squelette ne respecte pas le réalisme botanique du réseau de nervures. Cela peut engendrer des problèmes lors de la déformation des nervures (déformations non réalistes) ou bien lors de l'assignation des couleurs et du vieillissement. De plus, cette méthode utilise aussi un contour fourni par l'utilisateur.



FIGURE 1.20 – Maillage et rendu résultats de la méthode de modélisation par squelette.

1.2.1.5 Surfaces implicites.

La méthode utilisant les contours implicites développés par [MHW92] permet de modéliser les feuilles composées. Ces feuilles sont constituées de plusieurs folioles (petites feuilles) sur une tige. Pour ce faire, un squelette des nervures primaires de la feuille est généré grâce à un L-Système.

Ce squelette sera utilisé pour former la surface implicite définissant les contours de la feuille. De plus, cette méthode propose d'agir sur le rayon d'influence le long du squelette, ceci permet de modéliser les formes du contour assez finement (fig. 1.21).

Avec le squelette généré via un L-Système[MHW92] peut modéliser la feuille à toutes les étapes de la croissance du L-Système.

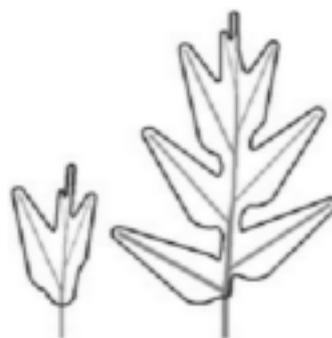


FIGURE 1.21 – Développement d'une feuille en contours implicites.

Cette méthode génère des contours implicites, suivant une fonction implicite $f : R^2 \rightarrow R$ avec $f(x, y) = c$, c étant une valeur seuil. Si $f(x, y) > c$ alors le point (x, y) est dans le contour, si $f(x, y) < c$ le point est hors du contour et enfin si $f(x, y) = c$ le point est sur le contour.

De plus, chaque segment généré par le L-système possède un facteur d'influence défini par la fonction suivante :

$$F(d) = \begin{cases} 2\frac{d^3}{r^3} - 3\frac{d^2}{r^2} + 1 & \text{si } 0 \leq d \leq r \\ 0.0 & \text{si } d > r \end{cases}$$

Avec d la distance entre un point et le segment considéré et r le rayon d'influence du segment.

Enfin, la fonction implicite est définie :

$$f(x, y) = \sum_{i=1}^n F(d_i(x, y))$$

Avec d_i la distance d'un point au i^{eme} segment.

Les segments ont un rayon d'influence variable sur leur longueur, pour ce faire un segment possède deux valeurs du rayon d'influence et une fonction permettant d'interpoler ces valeurs sur la longueur du segment (Fig 1.22).

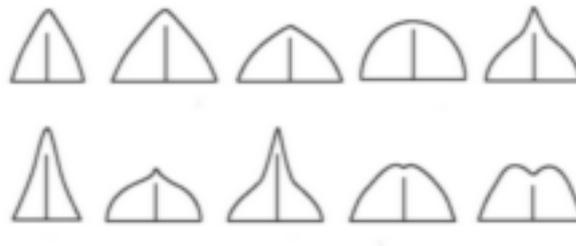


FIGURE 1.22 – Différents rayons d'influence interpolés.

Le contour est ensuite tracé en utilisant une grille régulière plane. Un point de départ est trouvé en prenant le milieu d'un segment et les points de la grille sont pris ou rejetés s'ils sont sur le contour ou non (Fig 1.23).

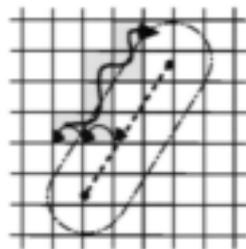


FIGURE 1.23 – Méthode de traçage du contour implicite.

Cette méthode permet de facilement définir les contours d'une feuille. Le processus, reposant sur un L-Système pour la création des nervures primaires, ne permet cependant pas de générer les nervures secondaires présentes dans le limbe des feuilles.

1.2.1.6 Discrete Shells.

[Bas06] propose de modéliser et de simuler les propriétés physiques des feuilles (fig. 1.24) en utilisant les "Discrete Shells" développées par [GHDS03]. Les "Discrete Shells" permettent de modéliser des structures fines et flexibles. Du fait de la nature géométrique des feuilles générées par cette méthode (facettes triangulaires) elles peuvent être facilement intégrées dans d'autres structures.

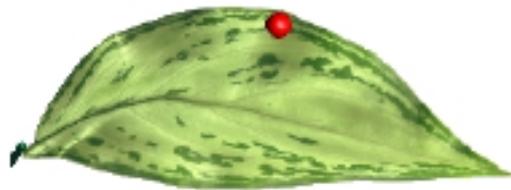


FIGURE 1.24 – Modélisation avec les Discrete Shells.

1.2.1.7 Système de particules.

Pour modéliser les nervures d'une feuille, la méthode de [YRL04] est basée sur l'utilisation d'un système de particules. L'utilisateur doit fournir une silhouette de la feuille ainsi qu'un point sur cette silhouette constituant la cible des particules. Les particules sont mises au hasard sur la feuille puis effectuent des déplacements vers la cible. Ces déplacements ont fait l'objet de deux algorithmes dans [YRL04]. Tout d'abord, les particules se déplacent toutes séparément en ligne droite vers la cible. Cela donne des nervures très peu réalistes (Fig 1.25).

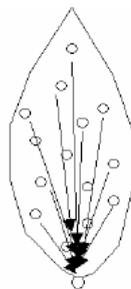


FIGURE 1.25 – Transport rectiligne des particules.

Dans une deuxième approche, [YRL04] propose de faire interagir les particules entre elles. De ce fait, les particules ne se déplacent plus en lignes droites, mais suivent un chemin en fonction de la cible et des autres particules. Pour ce faire, une énergie est affectée à chaque particule, c'est cette énergie qui permettra de connaître la largeur de la nervure. De plus si deux particules sont trop proches l'une de l'autre, elles sont alors fusionnées, et leurs énergies ajoutées pour former

l'énergie de la nouvelle particule. Le mouvement est donné par l'équation suivante :

$$v = \frac{(w_p v_1 + w_q v_2)}{w_p + w_q}$$

Avec :

v_1 la norme du vecteur particule->cible,

v_2 la norme du vecteur particule->particule la plus proche,

w_p le facteur de largeur de la nervure issue de la particule,

w_q le facteur de largeur de la nervure issue de la particule la plus proche.

La particule avance ensuite suivant ce vecteur d'un pas fixe (Fig. 1.26).

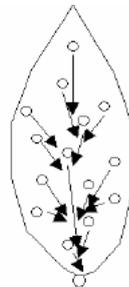


FIGURE 1.26 – Transport dépendant des autres particules.

Nous pouvons voir à la Figure 1.27 les différentes phases de développement de cette méthode.



FIGURE 1.27 – Développement du réseau de nervures.

1.2.1.8 Conclusion

Les méthodes présentées dans cette partie décrivent souvent des résultats éloignés de la réalité botanique des nervures de feuilles. Cependant les méthodes basées sur des données biologiques comme [RFL⁺05] ou sur des particules comme [YRL04] permettent d'avoir des nervures réalistes et basées sur des données et processus inspirés de la biologie.

Il est souvent nécessaire dans les méthodes présentées de fournir un gabarit donnant la silhouette de la feuille. Une telle silhouette permet de délimiter facilement le cadre de croissance et la forme du contour d'une feuille. Cependant, la génération d'un grand nombre de feuilles appartenant à une même espèce s'avère problématique. En effet, cette silhouette définit de façon très rigide la forme finale de la feuille et n'autorise presque aucune liberté sur un autre aspect que son réseau de nervure.

1.2.2 Rendu de feuilles

Le rendu de feuilles d'arbres est un problème complexe en synthèse d'image, en effet la structure en couches des feuilles et l'évolution biologique des pigments colorés lors du vieillissement sont difficiles à modéliser. Nous verrons dans un premier temps les solutions mises en oeuvre pour obtenir le rendu le plus réaliste possible de feuilles puis nous présenterons les méthodes visant à effectuer le vieillissement des feuilles.

L'étude des propriétés d'absorption, de réflexion et de transmission de la lumière a été réalisée par [BLD⁺06] sur une feuille de hêtre (Fig 1.28). Cette étude a permis de proposer une série de paramètres pour des BRDF et des BTDF applicables aux feuilles.



FIGURE 1.28 – Coupe transversale d'une feuille de hêtre.

Les mesures établies (Fig 1.29) permettent de simuler le rendu de feuille de hêtre. Elles sont exprimées en sr^{-1} .

1.2.2.1 Rendu sous surfacique.

Une feuille est composée de différentes couches de cellules. Tout d'abord, les premières couches protégeant la feuille de l'extérieur sont ses épidermes. L'épiderme adaxiale sur la face supérieure et l'épiderme abaxiale sur la face inférieure.

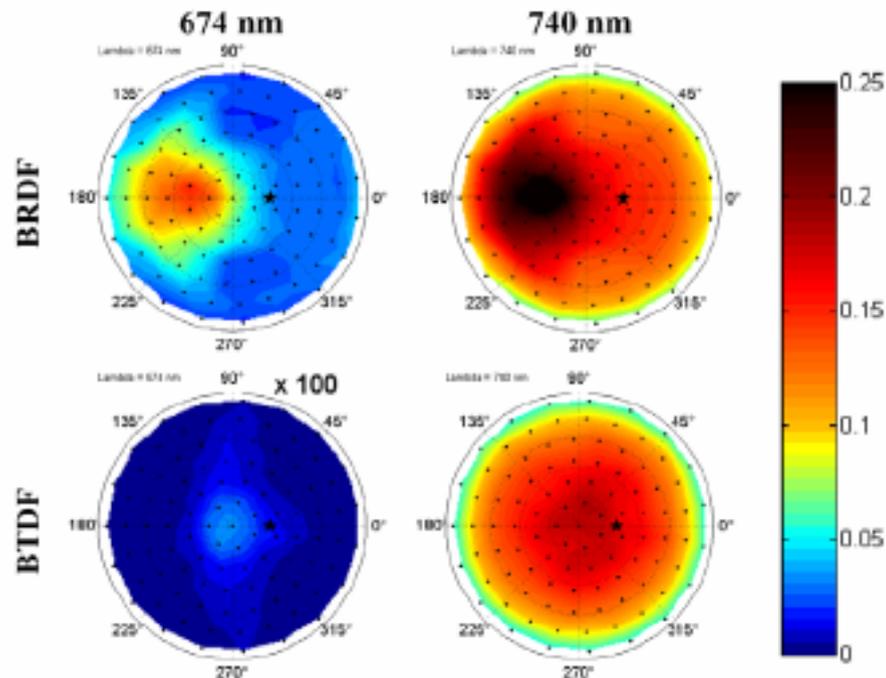


FIGURE 1.29 – reflectance et transmittance de la lumière sur une feuille de hêtre.

Ces deux couches sont composées de cellules dont les faces sont tournées vers l'extérieur et sont recouvertes d'une cuticule.

Entre ces deux couches protectrices, la feuille stocke les cellules qui lui permettront de réaliser la photosynthèse. Ces cellules sont appelées tissus du mésophylle (littéralement milieu de la feuille) et sont comprises entre les deux couches épidermiques de la feuille. Ce tissu au centre de la feuille est composé de deux couches, le parenchyme palissadique et le parenchyme spongieux. Le parenchyme palissadique, proche de la face supérieure, contient une importante quantité de chloroplastes visant à effectuer la photosynthèse. Le parenchyme spongieux agit quant à lui comme une interface pour échanger les gaz produits par la photosynthèse avec l'extérieur.

Du fait de la structure en couche d'une feuille, le rendu sous surfacique est la méthode la plus intuitive pour rendre des feuilles d'arbres. En effet, cette méthode permet de faire transiter la lumière suivant un modèle 3D en plusieurs couches et donc de produire un résultat biologiquement réaliste.

Modèle algorithmique : ABM.

Dans l'optique d'avoir un rendu réaliste de feuilles [BR97] proposent un modèle de réflectance et de transmittance algorithmique (ABM). L'ABM est l'acronyme de "Algorithmic BSSDF Model", la BSSDF (*Bidirectional Surface Scattering Distri-*

bution Function) étant la fusion de la BRDF (*Bidirectional Reflection Distribution Function*) et de la BTDF (*Bidirectional Transmission Distribution Function*).

Pour déterminer le trajet de la lumière au sein de la feuille, l'ABM utilise un modèle composé de couches (Fig. 1.30).

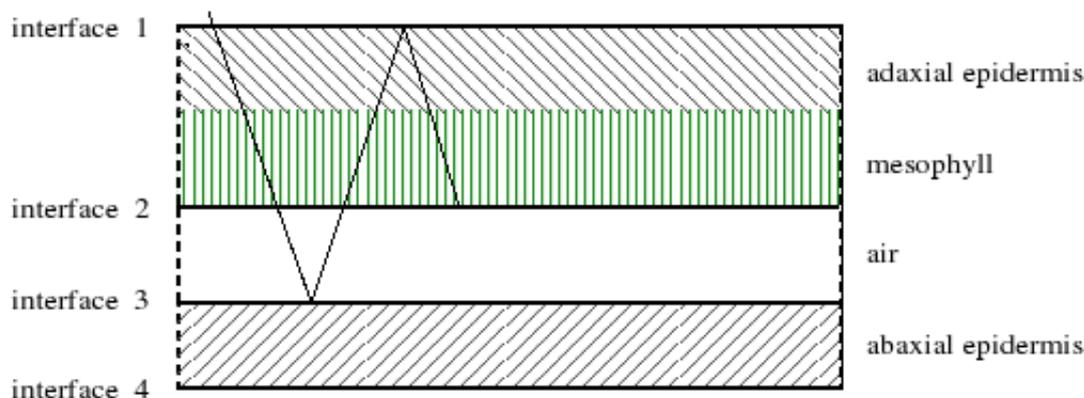


FIGURE 1.30 – Les différentes couches utilisées dans le modèle de [BR97] et un trajet possible d'un rayon de lumière.

Les interfaces entre l'air et les couches adaxiales et abaxiales (interface 1, 3 et 4) sont semi-réfléchissantes, l'interface entre l'air et le mésophylle (interface 2) est quant à elle semi-absorbante. Un rayon de lumière frappant la feuille peut, à l'interface 1 et 4, être réfléchi dans son environnement d'arrivée ou bien être réfracté dans le milieu cible. De ce fait, un rayon de lumière peut être "piégé" à l'intérieur de la feuille avant d'être absorbé par l'interface 2 (couche mésophylle) (Fig. 1.30).

Pour obtenir les coefficients de réflexion et de réfraction, l'ABM utilise les équations de Fresnel. Le trajet d'un rayon de lumière peut être assimilé à un déplacement aléatoire en considérant chaque probabilité de transition comme étant un coefficient de Fresnel. Cependant, les équations de Fresnel étant vraies pour une surface plane infinie, [BR97] prend comme simplification qu'une cellule de feuille est suffisamment plane et le processus suffisamment petit pour appliquer ce traitement.

Une fois ce coefficient calculé, un nombre aléatoire (compris entre zéro et un) est généré, il servira comme paramètre de décision pour savoir si le rayon est réfléchi ou s'il est réfracté. Pour décider d'un tel cas, le nombre généré est comparé au coefficient calculé, s'il est inférieur, le rayon est réfléchi, sinon le rayon est réfracté. La réfraction et la réflexion sont calculées avec la classique loi de Snell-Descartes.

Ce modèle offre une approche réaliste de l'interaction entre la lumière et un tissu végétal. En effet, grâce à l'utilisation de différentes couches représentant les strates biologiques retrouvées dans une feuille, Baranoski *et al.* peuvent proposer des résultats biologiquement cohérents de feuilles utilisant le rendu sous

surfacique. Ce modèle sera réutilisé dans le modèle foliaire de dispersion présenté dans [BR01].

Modèle foliaire de dispersion (FSM).

[BR01] Propose un moteur pour effectuer le rendu de plantes utilisant le modèle ABM cité précédemment. Baranoski *et al.* partent du constat que la surface des feuilles n'est pas totalement plane. En effet, cette surface comporte des micro détails même pour les feuilles cirées paraissant complètement lisses. Une perturbation est donc appliquée aux rayons de lumière lancés au travers des différentes couches utilisées par le modèle ABM. La perturbation est calculée de différentes façons suivant le type de l'interface prise en compte (absorption ou réflexion/réfraction).

Pour la réflexion et la réfraction, deux fonctions perturbatrices sont utilisées :

$$(\alpha_e, \beta_e) = (\arccos((1 - \xi_1)^{\frac{1}{ob+1}}), 2\pi\xi_2) \quad (1)$$

$$(\alpha_m, \beta_m) = (\arccos(\sqrt{1 - \xi_3}), 2\pi\xi_4) \quad (2)$$

Avec :

α_e et α_m : angles polaires respectant la réflexion ou réfraction du rayon propagé.

β_e et β_m : angle autour de la direction de propagation.

$\xi_{1,2,3,4}$: nombres aléatoires entre zéro et un.

La fonction (1) est utilisée lorsque le rayon arrive d'un milieu aérien et la fonction (2) lorsque le rayon a traversé la couche mésophylle.

Nous pouvons voir à la figure 1.31 les différentes perturbations en fonction des interfaces.

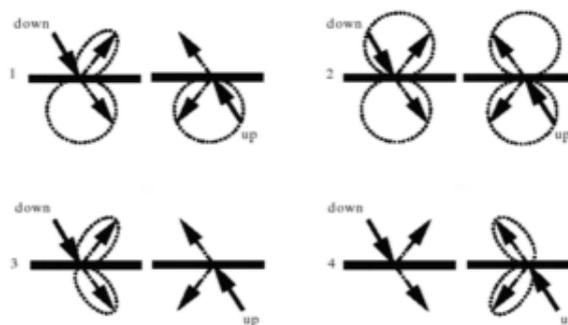


FIGURE 1.31 – Perturbations du rayon propagé.

L'absorption d'un rayon est ensuite déterminée lors de son arrivée dans la couche mésophylle. Suivant la concentration en pigments, le rayon a plus ou moins

de chances d'être absorbé puis envoyé sur l'interface suivante. Nous pouvons voir à la figure 1.32 les résultats de ce modèle.



FIGURE 1.32 – Résultats du rendu avec le FSM. Une source de lumière éclaire les feuilles par l'arrière ce qui laisse apparaître le réseau de nervures.

Cette méthode, très gourmande en temps de calcul, est biologiquement réaliste. Le principal inconvénient est d'obtenir de bons résultats pour un coup de calcul moindre, en effet le ratio temps/réalisme est assez difficile à régler.

Rendu temps réel.

[WWD⁺05] ont mis au point une méthode de rendu sous surfacique temps réel. Elle est basée sur un modèle de feuille paramétrique en tranche, comportant une épaisseur variable (Fig. 1.33).

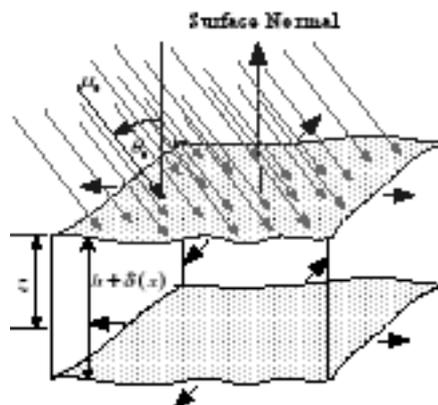


FIGURE 1.33 – Modèle en tranche d'une feuille.

Ce modèle inclut des variations d'épaisseur ($h + \delta(x)$) et une carte d'albédo $\gamma(x)$ (rapport de l'énergie solaire réfléchi par une surface sur l'énergie solaire incidente) pour amener un rendu non uniforme à la feuille.

La méthode utilisée pour rendre la feuille est basée sur une BRDF et une BTDF (f_r et f_t). La propagation sous surfacique de cette méthode est assurée par le modèle LEAFMOD qui a été validé expérimentalement. De plus, cette méthode inclut le rendu de surfaces rugueuses pour accroître le réalisme.

Le point fort de la méthode de [WWD⁺05] repose sur le modèle paramétrique, en effet les valeurs des épaisseurs et des albédos peuvent être récupérés à partir de vraies feuilles. En outre, le rendu est assuré par un algorithme d'illumination globale en deux passes permettant de générer des ombres réalistes.

Les résultats de cette méthode sont présentés à la figure 1.34.



FIGURE 1.34 – Résultats de la méthode de [WWD⁺05].

Les principales restrictions de cette méthode sont l'absence d'un modèle géométrique de feuille et la nécessité d'avoir les informations d'épaisseur et d'albédo pour les feuilles que l'on veut rendre. De plus, toutes les feuilles sont complètement similaires au niveau de leurs formes et de leurs textures pour chaque plante. Cependant, les résultats sont très réalistes.

1.2.2.2 Génération de texture avec un système de particules.

[YRL04] proposent, en plus de la génération de nervures grâce à un système de particules, de générer les nervures des feuilles avec un système de particules, de générer aussi les couleurs de la feuille. Pour ce faire, ils utilisent un champ de vecteurs modifié avec le parcours des particules (Fig. 1.35). Ce champ de vecteurs couvre toute la surface de la feuille et les vecteurs prennent la direction du sens de déplacement des particules, la taille des vecteurs indiquant la distance de la particule à ce vecteur.

Les couleurs sont ensuite insérées dans le champ de vecteurs. Celles-ci sont initialisées au niveau du pétiole et sont dispersées dans le champ en suivant la direction inverse des vecteurs (Fig. 1.36).

Quand la couleur se propage, un mélange est calculé. Ce mélange tient compte

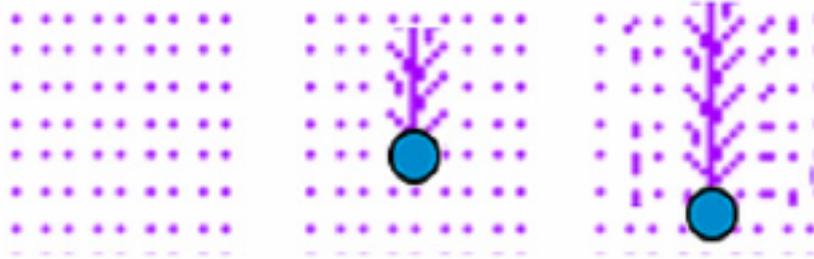


FIGURE 1.35 – Modification du champ de vecteurs avec le mouvement des particules.

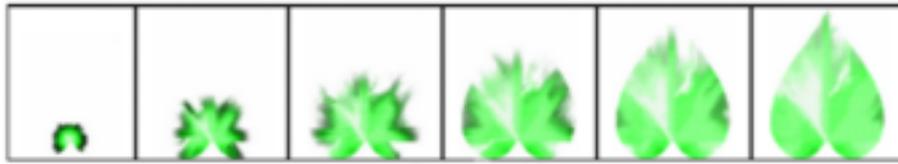


FIGURE 1.36 – Dispersion des couleurs dans la texture.

d'un poids de 0.7 pour la couleur déjà sur le champ de vecteur et de 0.3 pour la couleur se propageant.

Les résultats de cette méthode possèdent des nervures réalistes grâce à la génération utilisant un système de particules. Cependant, la génération de couleurs paraît peu convaincante étant basée sur une dispersion et non sur des éléments botaniques (Fig. 1.37).



FIGURE 1.37 – Résultats du rendu de [YRL04] avec une plante grimpante

1.2.2.3 Evolution et vieillissement.

Le vieillissement des feuilles est fonction de nombreux phénomènes biologiques. L'ensoleillement moyen en hiver étant un tiers de fois moins intense qu'en été environ, les feuilles commencent à changer. A l'origine de ce changement se trouvent des cellules, les phytochromes, qui détectent ce changement d'ensoleillement et activent le processus de sénescence des feuilles.

Une fois ce processus enclenché, la feuille commence par produire de l'éthylène. C'est ce gaz qui va s'attaquer à la chlorophylle et la détruire. Sans chlorophylle les feuilles perdent leur teinte verte et laissent la place à d'autres pigments naturellement présents dans la feuille : les anthocyanes et les caroténoïdes. Ces pigments donnent aux feuilles une couleur du rouge au jaune en passant par l'orange, mais plus important, ils permettent de protéger la feuille des températures froides le temps que l'arbre finit de rapatrier les nutriments dans son tronc.

Ensuite, différentes enzymes sont produites pour dégrader la couche de cellule qui retient la feuille sur l'arbre. Une fois que ces enzymes ont dégradé cette couche, une substance est sécrétée pour cicatrifier la plaie et effectuer un bouchon contre les possibles attaques extérieures. La feuille n'est plus retenue que par les anciens vaisseaux de sève et tombera à la moindre contrainte exercée par le vent ou tout autre phénomène extérieur.

[NCN96] définissent le vieillissement complet d'une feuille en automne en fonction des pigments contenus dans cette feuille. Les facteurs pris en compte pour l'évolution de la concentration de ces pigments sont l'éclairement, la distance d'une nervure au pétiole et le temps. La couleur est ensuite déterminée en fonction de ces différentes concentrations.

Une méthode pour obtenir le vieillissement des couleurs des feuilles en automne a été développée par [MCK⁺01]. Leur modèle se base sur l'éclairement solaire et des informations expérimentales de colorations. Les principales étapes de colorations de leur modèle dépendent de la concentration de pigments. Ils utilisent de plus des algorithmes chaotiques et de fluctuation pour introduire des variations dans les couleurs.

Une autre méthode pour simuler la coloration des feuilles à partir de concentrations de pigments est celle de [BDE04]. Leur algorithme détermine la quantité de lumière arrivant sur la feuille et en fonction de la quantité de pigment présente dans la feuille, trouve la couleur à appliquer.

1.2.2.4 Conclusion

La rendu de feuille comporte de multiples facettes. Tout d'abord, nous avons vu avec le modèle "ABM" de [BR97] utilisé dans [BR01] que le rendu de feuille peut être abordé d'une manière sous surfacique. En effet, les feuilles possèdent

une structure en couches particulièrement adaptée à ce traitement. Cependant, le temps de calcul mise en oeuvre est trop important pour justifier de son utilisation.

Le modèle de rendu sous-surfaciques vu dans [WWD⁺05] permet quant à lui d'utiliser des outils pour effectuer la visualisation des feuilles en temps réel. Leur modèle paramétrique pouvant intégrer des valeurs issues de vraies feuilles augmentant considérablement le réalisme. Cette méthode ne repose malheureusement pas sur les détails géométriques de la feuille, mais uniquement sur des valeurs acquises sur un modèle réel. Rodkaew *et al.* [YRL04] propose la génération automatique de textures pour une feuille. Bien que les nervures paraissent réalistes, les couleurs appliquées aux feuilles le semblent moins. Le mélange des couleurs semble artificiel et ne tient que très peu compte des nervures.

Le vieillissement des feuilles est quant à lui traité avec 3 principaux paramètres : la concentration de pigments, la distance d'un point par rapport aux nervures et l'ensoleillement. Ces paramètres seront repris dans notre modèle et augmentés grâce à une phase de vieillissement géométrique.

1.3 A une échelle supérieure : le choix des altérations de terrains

1.3.1 Altérations de terrains

La génération de terrains à petite échelle est un sujet relativement peu traité. Les domaines de recherche comme les terrains fractals et leur érosion ainsi que les caractéristiques géologiques des sols sont largement plus présents dans la littérature. La structure des sols influe de manière très significative sur leur comportement et leur évolution.

1.3.1.1 Érosion

L'érosion est un phénomène fondamental pour la génération et la dégradation de terrains. Elle influe sur l'aspect global d'un sol en dégradant les roches et les couches supérieures du sol et en faisant migrer les débris résultant vers d'autres zones du terrain. Les phénomènes responsables de l'érosion sont le plus souvent l'eau, le vent, la glace ou les vagues. Parmi ces phénomènes, l'eau est la principale cause d'érosion. En effet, l'eau courante possède une capacité d'érosion remarquable. Que ce soit le lit d'une rivière ou bien un sol soumis à la pluie, l'eau casse, creuse, dissout et transporte les éléments de ces terrains pour en remodeler complètement le visage.

Musgrave *et al.* dans [MKM89] ont proposé l'érosion de terrains fractals grâce à deux méthodes. La première se base sur l'érosion hydraulique causée par l'eau

courante. L'eau est "déposée" sur chaque sommet du terrain et transite ensuite vers des sommets se trouvant à une altitude inférieure par rapport au sommet considéré. Cette eau permet d'éroder le terrain en prenant en compte le volume d'eau et de sa concentration en sédiments. Puis, Les sédiments en suspension dans l'eau sont déposés à la surface du sol.

La deuxième méthode permet d'éroder un terrain grâce à la température. En effet, sous l'action des chocs thermiques de petits morceaux de sol se détachent et tombent en bas des pentes du relief. L'érosion décrite par Musgrave dans [MKM89] permet de vieillir un terrain généré avec une méthode fractale. Cependant, aucune contrainte géologique ne peut être imposée à cause du modèle, cette érosion peut donc être appliquée uniquement à des zones de terrain relativement uniformes.

L'érosion hydraulique est traitée de façon différente par Chiba *et al.* dans [CMF98], ils se proposent de créer un terrain comportant des crêtes et des vallées à partir de cartes d'élévations. Pour ce faire, ils utilisent le flux d'eau calculé à partir du mouvement de particules indépendantes, ce flux étant dépendant de la topographie du terrain. Chaque particule d'eau peut entrer en collision avec le sol et arracher un morceau de matière, au cours des itérations le relief érodé se dessine (fig. 1.38).

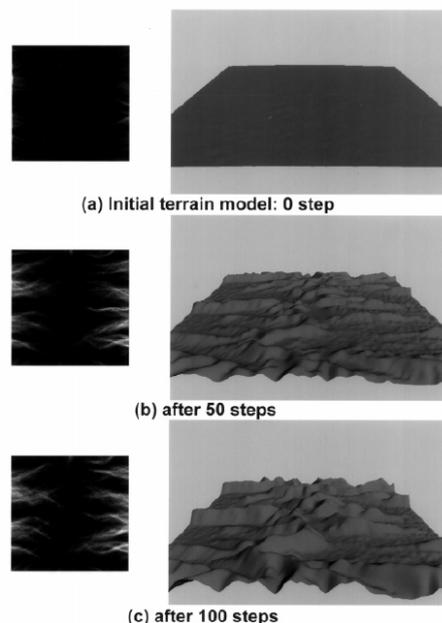


FIGURE 1.38 – Modification du relief au cours des itérations ([CMF98]).

De plus, Chiba *et al.* peuvent diriger leur modèle d'érosion grâce à trois paramètres permettant de régler la dureté du matériau utilisé pour le sol, le taux de transport des sédiments par l'eau et la puissance de la sédimentation.

Ce modèle permet de créer des paysages assez complexes et "physiquement" réalistes pour un coup de calcul moindre (méthode basée sur les cartes de hauteurs). Cependant, aucun paramètre ne permet de gérer l'aspect géologique d'un sol comportant plusieurs matériaux ou des matériaux hétérogènes. Cette méthode

peut être très intéressante pour la simulation d'érosion à petite échelle, car on y retrouve souvent le cas de dégradation d'un seul type de matériaux. Il reste tout de même le cas des objets posés sur le sol (cailloux, branches, feuilles, herbe) que cette méthode ne parviendrait pas à traiter, il faut un modèle plus évolué.

Un autre modèle de terrain présenté par Benes *et al.* dans [BF01] se base sur des aspects géologiques. Ce modèle est organisé sous forme de couches de matériaux (fig. 1.39) et se propose de réutiliser les méthodes d'érosion de Musgrave vues dans [MKM89].

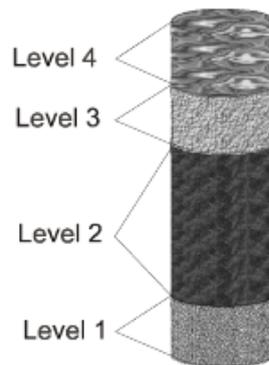


FIGURE 1.39 – Le modèle en couche de Benes ([BF01]).

Avec un tel modèle Benes *et al.* peuvent produire une érosion beaucoup plus réaliste géologiquement, les différentes couches possèdent leurs propres paramètres d'érosion et donc se dégradent indépendamment (fig. 1.40).

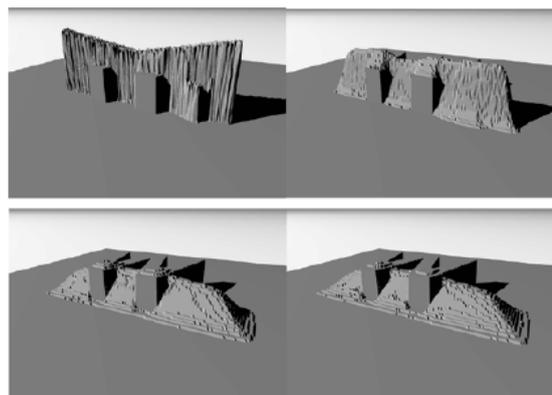


FIGURE 1.40 – L'érosion des différentes couches est indépendante. Ici le "W" est fait d'une matière très dure et est recouvert de boue.

Ce modèle possède l'avantage de disposer d'un aspect géologique pouvant intégrer des données réelles et donc de retranscrire l'érosion d'un sol existant (fig. 1.41). Cependant, ces données géologiques étant stockées dans des tableaux 2D, la visualisation des résultats donnent un aspect grossier laissant apparaître la structure sous-jacente au modèle. Ce défaut peut être résolu, mais amène un surcoût de calcul non négligeable.

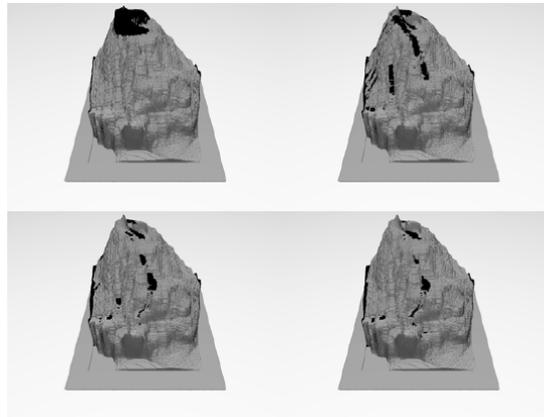


FIGURE 1.41 – L'érosion d'une montagne réelle, l'eau courante apparaissent en noir.

Benes *et al.* propose ensuite dans [BF02] un nouvel algorithme pour l'érosion hydraulique de sols. Cet algorithme est basé sur la dissolution de matériaux par l'eau et utilise quatre étapes indépendantes :

- **Injection d'eau dans le modèle** : De l'eau est introduite sous forme de sources ou de pluies. Les sources sont des zones localisées émettant un flux constant d'eau sur un seul élément, les pluies sont quant à elles dispersées sur de plus grandes étendues. Cette eau peut ensuite s'évaporer, ce phénomène dépend de la température et du niveau d'eau.
- **Érosion et captage de sédiments** : L'eau érode le sol par dissolution (fig. 1.42) et capte les sédiments au cours du temps.

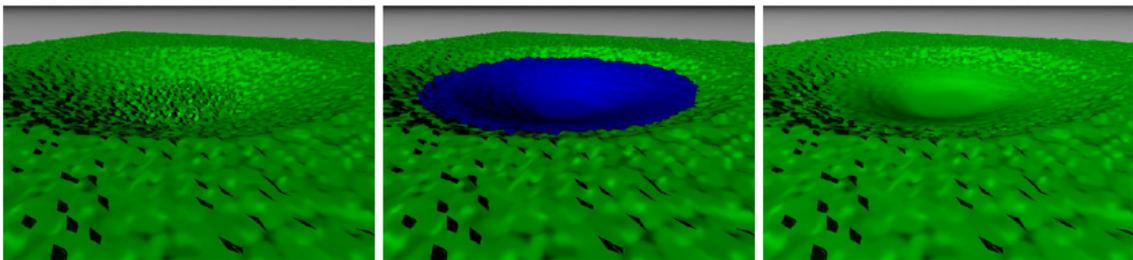


FIGURE 1.42 – Dissolution du sol par l'eau.

- **Transport de l'eau et des sédiments** : L'eau est transportée de proche en proche dans le modèle selon la hauteur d'eau sur chacun des éléments (fig 1.43.)
- **Dépôt de sédiments** : L'eau possède un coefficient de saturation, lorsque l'eau s'évapore le niveau de saturation est atteint et des sédiments sont déposés sur le fond pour revenir à la saturation maximale du volume d'eau après évaporation. La figure 1.42 montre ce dépôt dans le fond d'un cratère. Cette méthode présente l'avantage de bien simuler l'action de l'eau sur l'éro-

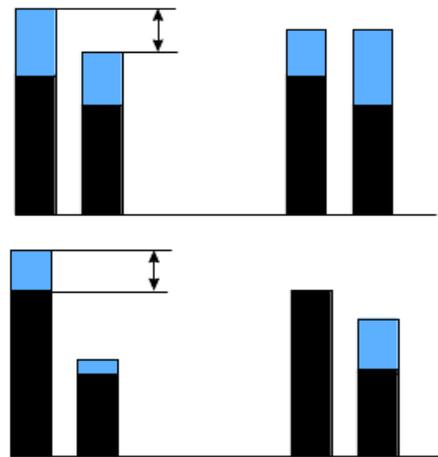


FIGURE 1.43 – Transport de l'eau et des sédiments dans les éléments du modèle (ici avec deux éléments). L'instant $t - 1$ se trouve à gauche et l'instant t à droite.

sion d'un sol et de dégager quatre traitements indépendants pouvant simuler tout type de climat (pluies, sécheresse ...). Cependant, la géologie du terrain n'est pas respectée et aucune donnée physique n'est prise en compte, de plus aucune gestion des particules de sédiment n'est utilisée ce qui amène un modèle très empirique.

A plus petite échelle, Valette *et al.* proposent de simuler la dégradation de sols par la pluie dans [VPLL06]. Leur méthode est basée sur un automate cellulaire pouvant prendre en compte l'épaisseur du sol ainsi que les échanges locaux entre cellules. Ces cellules sont des divisions régulières (ici un cube de 2mm de côté) possédant un grand nombre de sous états. Le volume d'une cellule est de plus composé d'un volume de gaz, d'un volume de liquide et d'un volume de solide, chacune des cellules de l'automate interagit avec ses voisines. Plutôt que d'utiliser une seule fonction de transition Valette *et al.* définissent une fonction par processus simulés, les fonctions de transition de l'automate utilisent différents voisinages en fonction de ces processus.

La pluie est la seule source d'érosion pour ce modèle et comporte deux aspects. Premièrement, les gouttes stressent le modèle et peuvent arracher des particules en surface pour les faire retomber suivant les éclaboussures. Deuxièmement, les sédiments peuvent être charriés ou non suivant leurs tailles et ainsi déposés sur le sol. L'eau courante charriant les sédiments vient de la pluie, de plus cette eau courante peut s'infiltrer dans le sol (fig. 1.44) en utilisant la saturation de ce sol en eau.

Enfin grâce à son automate cellulaire, le modèle de Valette *et al.* est très adaptable à différents types de sols. Cependant, il ne simule que la croûte du sol et ne permet pas de montrer l'aspect géologique d'un terrain.

Peytavie *et al.* dans [PGMG09] proposent un modèle hybride de génération et d'édition de terrains. Leur méthode permet de gérer des terrains volumiques pour un moindre coût mémoire grâce à l'utilisation de couches de matériaux. En plus

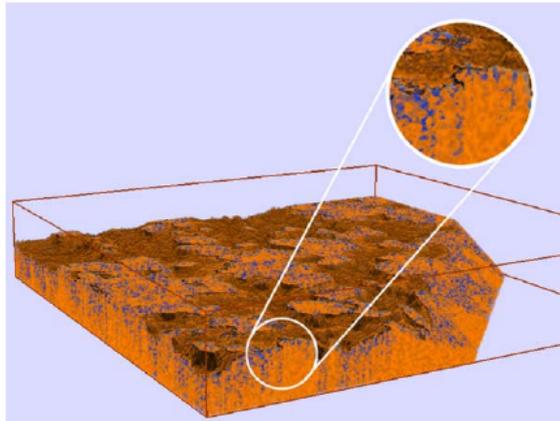


FIGURE 1.44 – Érosion et infiltration dans un sol érodé par la pluie.

des outils d'édition mis en oeuvre, leur modèle permet de générer de l'érosion qui engendrera des éboulements. Ces éboulements sont mis en place avec un processus d'empilement de rochers sur le sol. Nous pouvons voir le résultat de la génération d'un terrain avec une topologie complexe comportant des arches et des falaises sur la figure 1.45.



FIGURE 1.45 – Génération d'arches et de falaises avec une érosion créant des amas de rochers aux pieds des conformations rocheuses.

Cette méthode permet de générer et d'éditer efficacement un terrain tout en ayant des outils puissants pour gérer son érosion. Cette érosion se traduit par des empilements de rochers et l'apparition de couches de sable. Cependant, ce modèle ne se préoccupe pas des traces que peuvent laisser ces rochers sur le terrain lors de leur chute.

1.3.1.2 Fissures

Des aspects importants des sols sont les fissures. En effet lorsque le sol sèche ou subit une contrainte importante des fissures se forment en fonction de son élas-

ticité et de sa granularité. Ces fissures sont abordées de deux façons différentes : les modèles géométriques et les modèles physiques.

1.3.1.2.1 Les modèles géométriques Ces modèles utilisent des algorithmes pour générer des fissures vraisemblables. Ils sont le plus souvent basés sur des méthodes stochastiques permettant de simuler la forme d'une fissure sans aucune vraisemblance physique.

Hirota *et al.* dans [HTK00] décrivent leur modèle de fissuration, celui-ci est un modèle stochastique permettant de reproduire des formes de fissures en 2D. Leur modèle ne peut cependant pas générer des fissures de différentes largeurs et reste en 2D donc les fissures ne comportent aucune profondeur.

Perrier *et al.* quant à eux utilisent la tessellation de Dirichlet pour avoir une fragmentation fractale dans [PMRDM96]. Cette fragmentation possède plusieurs niveaux de détails pour simuler un sol boueux (fig. 1.46). Ici les fissures possèdent différentes largeurs, mais restent toujours en 2D et proposent aussi une structure fermée.

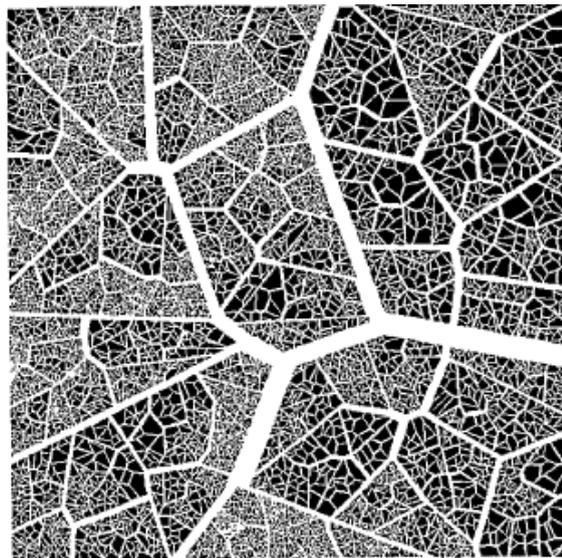


FIGURE 1.46 – Utilisation de la tessellation de Dirichlet sur plusieurs niveaux dans [PMRDM96]

1.3.1.2.2 Les modèles physiques Ils se basent sur la dynamique pour créer des fissures réalistes, mais sont beaucoup plus lents que les modèles géométriques.

Valette *et al.* dans [VPLL08] présentent un modèle de fissuration basé sur un automate cellulaire. Celui-ci est initialisé à l'aide d'une carte de hauteurs du terrain et d'une carte des épaisseurs de la couche superficielle du sol. L'automate est ensuite altéré en suivant le graphe des chemins pouvant être suivis par les fissures. Ce graphe est généré à partir d'une méthode classique comme la tessellation

de Dirichlet ou la ligne de partage des eaux. Le résultat obtenu en sortie est une surface triangulée fissurée (fig. 1.47).

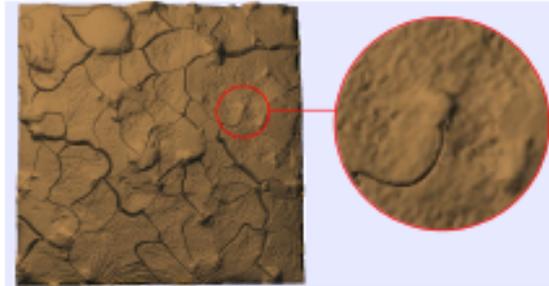


FIGURE 1.47 – Résultats de la méthode de Valette *et al.* [VPLL08].

En plus de traiter les sols, Valette *et al.* généralisent leur méthode aux fissurations de maillages 3D. Pour ce faire, ils paramétrisent la surface 3D et interpolent les déplacements des fissures sur cette paramétrisation.

Leur méthode possède l'avantage d'être 3D et donc de pouvoir générer les fissures dans leur profondeur. De plus, c'est une méthode basée sur la physique et il suffit de fournir peu de données (carte de hauteur du terrain et carte d'épaisseur de la couche superficielle) pour obtenir les résultats. Cette méthode reste cependant assez lente, il faut plusieurs dizaines de minutes pour calculer le maillage résultant comportant les fissures.

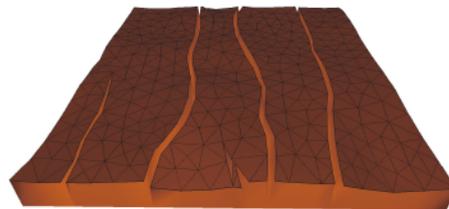


FIGURE 1.48 – Fissures en utilisant la mécanique du solide et les éléments finis

Federl dans le cadre de sa thèse ([Fed02]) utilise un système de ressorts présenté par Skjeltorp dans [Skj88] pour simuler les contraintes appliquées à un terrain. Cependant, l'inclusion de la croissance des fissures grâce à ce système de ressorts l'a amené à utiliser plutôt une méthode basée sur la mécanique du solide et les éléments finis (fig. 1.48).

1.3.1.3 Altération du sol par des phénomènes extérieurs

Les enfoncements d'un sol peuvent être de différentes sortes. Nous pouvons observer des traces dues à la chute d'un objet, des empreintes déposées par un marcheur ou bien un animal ou encore des ornières et traces laissées par un véhicule. Deux grandes familles de méthodes répondent à ce problème en synthèse d'image : les modèles physiques et les modèles empiriques.

1.3.1.3.1 Modèles physiques Que ce soit pour simuler les empreintes d'un marcheur ou bien pour modéliser une avalanche ou des éboulis, les modèles physiques permettent d'avoir une simulation physiquement réaliste des déformations du sol. Cependant, la majorité de ces méthodes demandent un temps de calcul élevé.

Chanclou *et al.* ([CLH96]) ont développé un modèle physique pour les sols mous marqués par des objets en mouvement. Ils décrivent les caractéristiques et les contraintes d'un sol mou sous deux principales facettes : la compression et le cisaillement.

Pour la contrainte de compression, les forces appliquées le sont longitudinalement. Le sol s'écrase et les espaces entre les différentes particules du sol sont fortement réduits et les particules du sol sont déplacées. Ces espaces sont pour la plus grande partie constitués d'air ou bien d'eau. Ils partent du constat que les particules ainsi déplacées font augmenter la résistance du sol par rapport à l'objet en mouvement d'une façon exponentielle. Une fois que les particules ne peuvent plus bouger, le sol est totalement comprimé.

Dans les contraintes de cisaillement, les couches de matières bougent les unes sur les autres. La résistance du sol résulte donc de deux phénomènes : la force de friction des particules ainsi que leur cohésion. La friction est proportionnelle au poids exercé perpendiculairement aux plans de cisaillement et la force de cohésion des particules est quant à elle une constante dépendant du type de matériaux présent. Par exemple, la cohésion d'un sable sec est de zéro alors que de l'argile possèdera un indice de cohésion élevé.

Enfin, leur étude du sol leur permet de mettre en avant trois principales déformations pour gérer les sols meubles. Premièrement, les petites déformations dans des zones élastiques où les matériaux reviennent à leur position initiale sans aucun dommage. Ensuite, les contraintes plus fortes pour lesquelles le matériau ne retrouvera pas sa forme initiale. Enfin, les forces extrêmes qui cassent le sol de manière irrémédiable et affectent grandement sa topologie.

La méthode proposée par Chanclou *et al.* repose sur l'interaction entre la déformation du sol et les mouvements de l'objet. En effet, les mouvements de l'objet affectent le sol et celui-ci affecte en retour les mouvements de l'objet. De plus, leur modèle propose de séparer les traitements pour les modifications à grande et petite échelle.

Un modèle de matériaux plastique est appliqué aux déformations de grande envergure. Il utilise des systèmes masse-ressort pour générer l'élasticité du sol et ses déformations. Ce réseau pour déformer le sol à grande échelle sera nommé "réseau de contrôle". Un modèle de matériaux granulaire est quant à lui utilisé pour les petites déformations du terrain et dans les déformations ayant besoin de raffinement. Ce "réseau de raffinement" recevra les déformations de l'objet en mouvement pour ensuite les transmettre au réseau de contrôle. Ces deux systèmes peuvent communiquer pour garder une cohérence des déformations à chaque échelle.

Cette méthode ne permet pas de modéliser et de gérer la structure complète du sol. En effet, seule une surface de déformation est gérée et aucune modification ne peut être répercutée sur une structure complexe de sol comportant différents matériaux empilés.

Le modèle physique décrit dans [LM93] par Li *et al.* vise à simuler les glissements de terrain grâce à une structure en piles de matériaux et l'interaction entre le sol et des engins de chantier. Leur modèle de sol se base sur des études géotechniques et sur la résistance au cisaillement du sol. Cette résistance est calculée en fonction de la friction définissant la résistance au glissement des particules composant le sol, de la cohésion et de l'adhésion entre ces particules et enfin des amas de particules solides résistants à la déformation.

Cette résistance au cisaillement est modifiée en fonction du type de sols considéré et des éléments climatiques extérieurs. En effet, le vent, la pluie l'humidité et la gravité influent sur cette valeur pour chaque type de sol. Cette résistance est ensuite utilisée pour gérer les échanges de matériaux entre les différentes couches du modèle. Chaque couche aura un point d'équilibre qui, une fois dépassé, fera s'écrouler le haut de la pile sur les piles voisines.

Ce modèle de sol permet à Li *et al.* de calculer l'interaction avec une lame de bulldozer. Un arc de cercle divisé en couches constitue une approximation de cette lame. Les contraintes seront calculées pour chaque couche et les échanges de matières interviendront entre ces couches. La deuxième application de ce modèle est l'interaction avec un godet. Comme pour la lame de bulldozer, le godet définira des tranches de sol où les échanges de matière auront lieu. Ces deux applications et les échanges de matière générés obéissent au modèle physique de sol et prennent en compte le point d'équilibre du matériau composant le sol. Nous pouvons voir sur la Figure 1.49 le mouvement effectué par la structure du sol lors du passage d'un bulldozer.



FIGURE 1.49 – Le sol est modifié par la lame d'un bulldozer avec la méthode de Li *et al.*.

La méthode proposée par Li *et al.* permet de faire interagir un modèle de sol granulaire avec des engins de chantier. Grâce à une approche physique, les lames et godets peuvent modifier le modèle de terrain en couches. Cette méthode ne permet cependant pas d'avoir un autre type d'objet interagissant avec le sol.

Sumner *et al.* ([SBH99]) introduisent un modèle physique pour l'interaction entre le sol et un objet en mouvement. Leur modèle de sol déformable repose sur une carte de hauteur supportée par des colonnes de matériaux. Ces différentes colonnes sont les cases d'une grille régulière discrétisant le sol. Les propriétés du modèle peuvent être modifiées pour permettre la simulation de différents types de matériaux où des objets rigides entreraient en impact.

Cette méthode utilise une grille régulière dont la résolution est fixée à l'initialisation et en concordance avec le modèle entrant en collision avec le sol. Cette grille prend ensuite les valeurs de la carte de hauteur pour chaque colonne de matière qui constituera la surface du terrain. Une série de passes d'érosion est ensuite appliquée au terrain pour lisser les différentes colonnes voisines et obtenir un résultat plus réaliste.

Sumner *et al.* ont implémenté plusieurs types d'interactions entre un objet rigide et leur modèle de sol déformable. Premièrement, la collision régit les phénomènes d'interaction directe entre un objet et le sol. Les colonnes qui prennent part à la collision sont tout d'abord déterminées et sont abaissées au niveau du point de collision.

Ensuite, le déplacement de matière est calculé et les quantités de matière abaissées pendant l'étape de collision sont distribuées aux colonnes adjacentes les plus basses. Le surplus de matière résultant de la collision sera donc redistribué dans les colonnes les plus proches de l'objet qui n'entrent pas en collision avec lui.

Enfin, un algorithme d'érosion est utilisé pour rendre le résultat plus naturel. Effectivement, les quantités de matière sous l'empreinte ayant été déplacées se voient toutes accumulées en bordure d'empreinte. Cet algorithme est aussi utilisé pour donner au matériau du sol différents comportements suivant les souhaits de l'utilisateur. La Figure 1.50 montre la formation d'une telle empreinte dans de la neige.



FIGURE 1.50 – Un piéton laisse une empreinte de chaussure dans la neige grâce à la méthode de Sumner *et al.*

La gestion des matériaux de Sumner *et al.* permet de produire des résultats réalistes d'empreintes sur un terrain. Leur méthode propose de traiter les sols meubles comme la boue, la neige et le sable. En effet, la réaction de ces trois matériaux reste assez similaire à traiter au regard du modèle physique utilisé. Les différentes passes d'érosions qu'ils utilisent représentent un frein à la création d'un grand nombre d'empreintes. De plus, l'utilisation d'une grille régulière possédant une résolution fixe ne permet pas de traiter de grandes étendues de terrain.

Cai *et al.* dans [CLS07] décrivent un modèle de terrain basé sur une grille régulière dépendant du point de vue. Un modèle physique est utilisé pour déformer les blocs du terrain. Ce modèle décrit les interactions entre un pneumatique et un sol en se basant sur le modèle de Bekker [Bek69] qui leur permet de procéder à une déformation physique de la structure de leur sol.

Une fois la détection de collision et l'abaissement physique du terrain effectués pour l'image courante, une texture procédurale du terrain est générée. Cette texture va permettre de mettre à jour les informations de chaque bloc de la grille. Ensuite, des bandes de polygones sont calculées perpendiculairement à la vue et sont utilisées comme masques pour stocker les différents niveaux de détails du terrain.

Enfin, l'image courante du terrain est calculée en utilisant les masques de bandes calculés en amont. Pour se faire, chaque bloc de la grille régulière est rendu suivant le niveau de détail du masque auquel il appartient. Nous pouvons voir Figure 1.51 les différents niveaux de détails ainsi que le résultat de cette méthode.

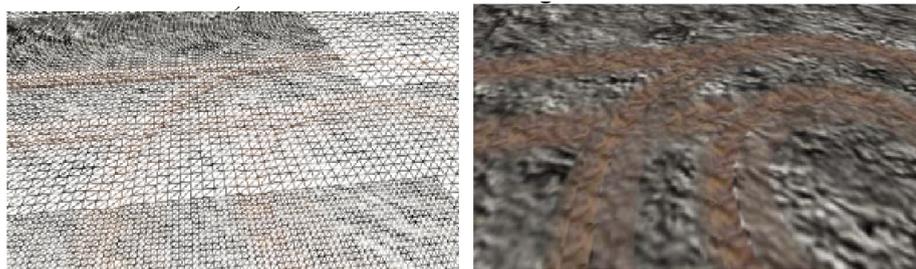


FIGURE 1.51 – Les différents masques fournissent les niveaux de détails du terrain (sur la gauche) et permettent d'obtenir le rendu final (sur la droite).

Avec une méthode alliant niveaux de détails et moteur physique, Cai *et al.* proposent un modèle de terrain pouvant supporter le passage de véhicules tout en étant déformé. La grille régulière sous-jacente est affectée par des niveaux de détails pour diminuer le nombre de polygones tout en gardant une résolution élevée au premier plan de l'image. Ce modèle est toute fois limité à l'impact de pneumatiques sur un sol meuble.

D'une façon plus générale, Zeng *et al.* proposent un modèle physique pour la déformation de matériaux granulaires interagissant avec des objets rigides ([ZTT⁺07]). Ce modèle repose sur un volume de matériau granulaire dont la surface est subdivisée suivant une grille uniforme. Cette grille est représentée par une carte de hauteur donnant à chaque case de la grille une hauteur bien définie. Comme pour les autres méthodes utilisant des grilles régulières et voulant initialiser la hauteur de chaque case, Zeng *et al.* peuvent le faire de différentes manières (carte de hauteur, outils de modélisation).

Tout d'abord, la collision avec l'objet en mouvement est détectée grâce à un lancer de rayon basique. Une fois l'objet entré en collision avec le sol, la matière ap-

partenant aux colonnes en dessous de l'objet doit être compactée. Pour ce faire, la matière est distribuée depuis les colonnes en contact avec l'objet vers les colonnes situées sur son contour. Ce déplacement de matière est géré physiquement par le modèle et permet de déformer la structure du sol sous la contrainte d'un corps rigide.

De plus, le mouvement de l'objet considéré est lui aussi affecté par le matériau granulaire. Nous pouvons observer sur la Figure 1.52 la chute d'un Rubik's cube et son impact laissé sur un sol composé de sable.

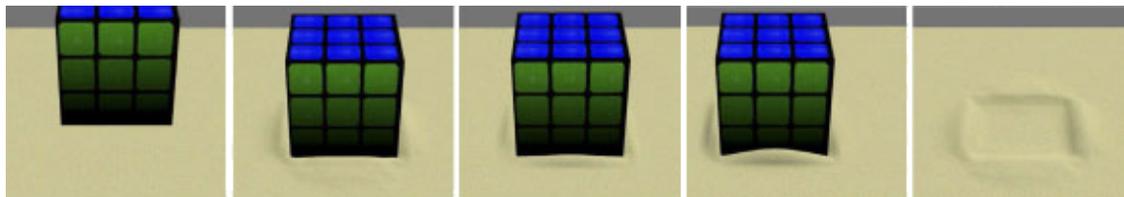


FIGURE 1.52 – La chute d'un Rubik's cube (de gauche à droite) laisse sur le sol une empreinte en forme de carré.

La réaction du terrain au comportement de l'objet animé est produite de façon réaliste grâce à l'approche physique de Zeng *et al.*. La limitation majeure de cette méthode est de ne pas pouvoir effectuer un grand nombre de traces sur le sol. En effet, comme les modèles physiques vus précédemment, ces simulations sont très efficaces pour générer les traces d'un objet bougeant localement sur le sol, mais ne peuvent que difficilement générer une grande collection de traces au sol.

1.3.1.3.2 Modèles empiriques Dans [ABZO06], Aquilio *et al.* proposent un modèle utilisant le GPU pour déformer dynamiquement les terrains à petite échelle. Leur méthode repose sur une carte de hauteur déplacée dynamiquement (DDHM pour Dynamically-Displaced Height Map) mise à jour à chaque pas de la simulation. Ces modifications permettent de garder les modifications topographiques tout au long de la simulation. De plus, chaque carte modifiée est enregistrée pour produire un historique de l'activité sur le terrain.

Premièrement, l'initialisation permet de transformer les données d'entrée de la carte de hauteur en DDHM. Ensuite les élévations et abaissements du terrain sont calculés en effectuant le rendu du tampon de profondeur dans une texture puis en utilisant un pixel shader pour déterminer la modification correcte de profondeur. Ces modifications sont bien évidemment effectuées en fonction de l'objet en mouvement dans la simulation.

La détection de collision entre le terrain et des objets rigides en mouvement est assurée par le moteur physique Tokamak [Lam87]. L'enfoncement est calculé en fonction de la position de l'objet puis permet de générer toute la simulation de déformation de terrain par un véhicule. Nous pouvons observer sur la Figure 1.53 le résultat de cette méthode.

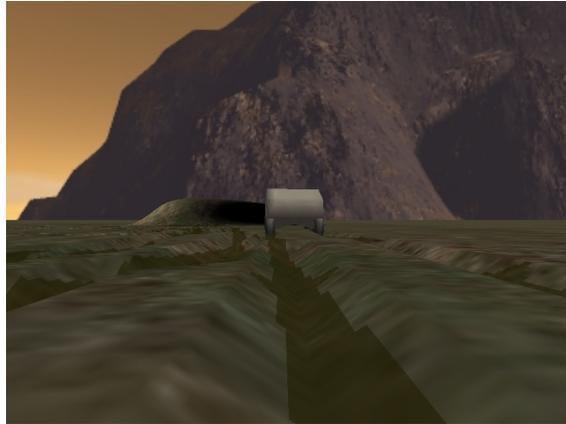


FIGURE 1.53 – Les ornières réalisées dans [ABZO06] en temps réel par un traitement sur GPU.

Bien qu'en temps réel, la méthode élaborée par Aquilio *et al.* utilise un modèle de terrain ne pouvant gérer que très peu de contraintes. L'inconvénient de ce modèle empirique est de ne gérer qu'un nombre limité de déformations. Ici, seules les traces linéaires produites par un véhicule peuvent être modélisées. De plus, aucun échange de matériaux n'est réalisé en dehors des enfoncements dus au véhicule.

Chen *et al.* développent un algorithme de visualisation d'un sol altérable par des pneumatiques dans [CZXY07]. Leur modèle de sol utilise un terrain dynamique multirésolution avec un graphe orienté acyclique. En effet, dans les simulations de véhicules, la résolution doit être beaucoup plus importante au niveau des empreintes.

Pour ce faire, Chen *et al.* utilisent une grille régulière initiale ayant des cases d'une taille de 1/64 m. Pour augmenter localement la résolution de la grille et modifier plus finement les cases de résolutions différentes, elles sont remplacées par des nouvelles ayant la résolution désirée. Cette grille est ensuite déformée par une simulation mécanique de roues évoluant sur le sol.

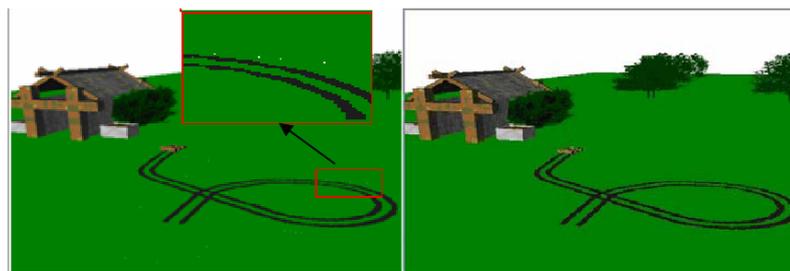


FIGURE 1.54 – Les traces de roues visualisées avec des niveaux de détails par [CZXY07]. Sur la gauche, sans zones de transition et sur la droite avec.

Du fait de la présence de nombreuse résolution, Chen *et al.* doivent maintenir un maillage cohérent. La notion de zones de transition est incluse dans le modèle. Ces zones auront pour charge de permettre la transition entre les différentes résolutions du maillage. En gardant une continuité entre les différents niveaux, ils

évitent les fissures engendrées par une telle méthode. Nous voyons sur la Figure 1.54 la création d'ornières sur un sol herbeux.

La méthode proposée par Chen *et al.* permet de visualiser rapidement des traces de roues effectuées dans un sol meuble. L'utilisation d'une grille régulière et d'une gestion des incohérences du maillage incorporées avec l'ajout de multirésolution s'approche d'un modèle topologique. Cependant, un modèle topologique comporte plus de cohérence au niveau des données et ne nécessite aucun traitement extérieur pour garder sa cohérence.

Chapitre 2

Génération de grandes quantités de feuilles à différents âges

Sommaire

2.1 Introduction	59
2.2 Problématique	60
2.3 Le 2-G-Carte L-Systeme paramétrique	61
2.3.1 Notations et outils	61
2.3.2 Génération de la forme de la feuille	62
2.4 Vieillessement de la feuille	80
2.4.1 Aspects géométriques du vieillissement	80
2.4.2 Vieillessement colorimétrique	83
2.5 Résultats	85
2.6 Conclusion	92

Chapitre 2

Génération de grandes quantités de feuilles à différents âges

2.1 Introduction

En nous basant sur l'étude des travaux antérieurs, nous pouvons constater que de nombreux résultats parmi les plus convaincants sont réalisés avec un L-Système. En effet, cet outil fournit une très bonne représentation des systèmes arborescents et ramifiés. Les feuilles végétales font naturellement partie de cette catégorie et peuvent facilement être décrites par un L-Système. Cependant, les L-Systèmes classiques ne permettent pas de générer la topologie d'une surface, dans le cas de feuilles cette topologie peut être complexe (sauf le cas des Map L-Systèmes d'un usage difficile).

Dans cette partie de nos travaux, nous avons plusieurs objectifs principaux. Tout d'abord la facilité de conception d'une feuille au niveau de sa forme. Ensuite, la possibilité de lui appliquer un vieillissement aussi bien colorimétrique que géométrique. Enfin, la capacité de générer une grande variété de feuilles pour une espèce et un âge donné.

Pour se faire, nous avons développé un modèle reposant sur un 2-G-Carte L-Système paramétrique. Cet outil, la fusion d'un L-Système et d'une 2-G-Carte, nous permet de facilement définir des grammaires décrivant la topologie d'une feuille. Nous avons travaillé sur un modèle global de feuille plutôt que sur des propriétés précises. Ainsi, notre modèle nous permet de traiter la génération de la forme d'une feuille ainsi que son vieillissement en utilisant la même structure. En effet, la forme de la feuille sera décrite par la grammaire du L-Système et sa topologie gérée par la 2-G-Carte.

Ces travaux ont été présentés lors de la conférence Computer Graphics International 2008 et publiés dans la revue "The Visual Computer" [PTMG08].

2.2 Problématique

Pour générer une feuille, on veut pouvoir gérer la croissance du réseau de nervures de cette feuille. Contrairement à [QTZ⁺06] ou bien [MMPP03], [YRL04] et [RFL⁺05] qui fournissent soit une image de feuille à reconstruire, soit une silhouette en 2 dimensions de celle-ci. Nous voulons que les feuilles générées par notre méthode soient entièrement définies par la forme de leur réseau de nervure.

Ce réseau correspond à une structure arborescente qui peut facilement être décrite par un L-Système comme ceux présentés dans [PL96]. Cependant, ces L-Systèmes classiques ne décrivent bien que les structures composées de mots ou bien de segments. Nos feuilles devront être composés de faces pour permettre un meilleur contrôle de leur forme et pour refléter au mieux la réalité botanique de ces organes. Pour ce faire, nous avons utilisé un L-Système non plus pour former un mot ou bien une arborescence de segments, mais pour façonner l'agencement des faces dans une 2-G-Carte. En effet, ce modèle présenté dans [Lie89] et [Lie94a] permet de gérer topologiquement des subdivisions en dimensions de l'espace. La réunion de ces deux outils, les L-Systèmes et les 2-G-Cartes, nous offre la possibilité de construire facilement n'importe quel agencement de faces 2D dans l'espace 3D. Le réseau de nervure ainsi créé sera rempli pour former tout le corps de la feuille.

Une autre facette que nous voulions aborder est le vieillissement des feuilles. Avec l'âge une feuille va complètement changer de forme lorsque ses nervures vont se flétrir puis sécher et s'enrouler sur elles même. Elle va aussi peu à peu changer de couleur pendant que la chlorophylle se dégrade dans son corps et laisse apparaître les autres pigments colorés. Le vieillissement géométrique d'une feuille sera assuré par la définition de la grammaire dans notre modèle. Avec les ajouts et améliorations apportés à celle-ci, les nervures pourront être courbées pour simuler une feuille flétrie.

Comme [NCN96], la distance aux nervures jouera un grand rôle pour le vieillissement colorimétrique d'une feuille dans notre modèle. Bien que les couleurs ne soient pas dispersées dans le corps de la feuille à partir des nervures comme dans [YRL04], les textures générées par notre modèle prendront en compte deux couleurs qui seront mélangées suivant la distance aux nervures. Un autre paramètre important du vieillissement est l'ensoleillement dans [MCK⁺01] et [BDE04]. Ce phénomène sera simulé en utilisant un masque de mélange dans notre modèle.

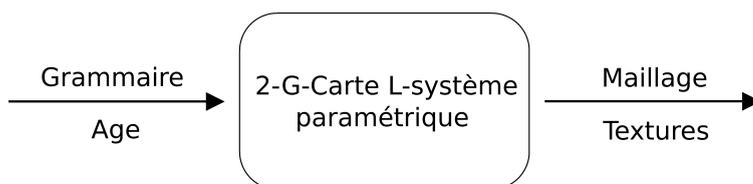


FIGURE 2.1 – Le principe de fonctionnement de notre modèle de génération de feuilles.

La figure 2.1 offre une vision du pipeline de notre méthode. L'utilisateur fournit une grammaire ainsi qu'un âge à donner à la feuille, ces deux paramètres sont ensuite utilisés pour le 2-G-Carte L-Système paramétrique. Le maillage final est composé en fonction de la grammaire, donnant le réseau de nervure à l'âge voulu ainsi que le limbe de la feuille. D'autre part, la texture de la feuille est définie en utilisant les paramètres botaniques de distance aux nervures et deux couleurs représentant deux âges différents. Ces textures peuvent ensuite être mélangées avec un masque pour produire des motifs plus complexes.

2.3 Le 2-G-Carte L-Systeme paramétrique

Notre modèle repose sur l'utilisation d'un 2-G-Carte L-Système. Comme pour les 3-G-Carte L-Systèmes développé dans [TGM⁺08] et [LTBG09], il nous offre la possibilité d'utiliser la flexibilité des L-Systèmes et la gestion des subdivisions en 2 dimensions de l'espace proposées par les 2-G-Cartes. La grammaire issue du L-Système comporte quelques changements par rapport à la version de base. En effet, nous avons rajouté des notations et des outils afin de faciliter l'écriture des grammaires. Dans un premier temps, nous présenterons ces outils puis nous reviendrons sur la génération proprement dite des feuilles.

2.3.1 Notations et outils

La première fonctionnalité rajoutée dans notre modèle est le nommage des faces et des arêtes. Dans notre modèle, chaque face sera caractérisée par un label (une lettre majuscule) qui servira de référence dans la grammaire. De plus, chaque arête reçoit aussi un label, cette fois de façon automatique, représentant sa position sur la face décrite. Une face comporte obligatoirement une arête origine notée "O" et une face extrémité noté "E". Les arêtes sur les cotés reçoivent quant à elles un label " C_x ".

Nous pouvons voir sur la Figure 2.2 le résultat d'une telle labélisation. L'arête origine "O" est située en bas, l'arête extrémité "E" en haut et les deux arêtes coté " C_1 " et " C_2 " de part et d'autre de la face. Ces différentes arêtes sont de plus identifiées grâce au label de la face. Par conséquent, les arêtes de la face A présente sur la gauche de la Figure 2.2 seront notées : A_O , A_{C_1} , A_{C_2} et A_E .

Ce sont ces notations qui seront utilisées pour simplifier l'écriture de la grammaire. Chaque type de face possédera un label permettant de l'utiliser dans les règles de production.

Nous avons ensuite besoin de pouvoir définir une face dans la grammaire. Cette déclaration est faite en utilisant un préambule spécial comportant :

- n : le nombre d'arêtes de la face.
- t : le coefficient de translation de la face
- α, β et γ : les angles d'orientation de la face.
- l : la longueur de la face.
- w : la largeur de la face.

De plus, nous donnons le label de la face dans sa définition. Au final, une face est définie dans la grammaire sur le modèle suivant :

#define label($n, t, \alpha, \beta, \gamma, l, w$)

Enfin, un axiome doit être spécifié pour servir de point de départ à la génération. Cet axiome sera la première face créée automatiquement dans notre modèle et servira de base aux autres faces. Les autres faces seront ajoutées au modèle en utilisant les règles de productions définies par la suite.

La grammaire sera l'ensemble des définitions des faces, l'axiome et les règles de production. L'utilisateur doit fixer un nombre d'itérations, c'est-à-dire le nombre de fois que les règles seront appliquées pour donner la forme finale de la feuille. Au cours de ces itérations, les règles sont appliquées de façon séquentielle. La première règle définie dans la grammaire est donc appliquée au début de chaque itération.

2.3.2 Génération de la forme de la feuille

La forme désirée pour la feuille sera donnée par l'interprétation des règles de production. Ces règles fourniront, à chaque étape de la génération, les faces à ajouter ou bien celles qui seront modifiées au modèle. Nous avons implémenté les règles classiques utilisées dans les L-Systèmes. Cependant, au contraire des L-Systèmes qui travaillent sur des segments de droite, nous travaillons maintenant avec des faces. Ce sont donc ces faces qui seront utilisées dans les règles de production pour façonner la forme des feuilles. Nous partons du constat qu'une feuille sera complètement définie par la forme de ces nervures et, de ce fait, par la grammaire définissant la croissance de ces nervures.

2.3.2.1 Définition des règles de construction

Les règles de productions peuvent être de différent type. Nous pouvons trouver des règles qui font croître le modèle en ajoutant des faces, d'autres qui collent les arêtes entre elles ou bien encore qui divisent une face en deux faces de label différent.

Croissance :

La première règle dont nous avons besoin est une règle pour ajouter une face au modèle en spécifiant sa position. Cette position sera l'arête d'une autre face que nous appellerons face mère. Une telle règle est définie comme suit :

$$A \rightarrow A[B]_E$$

Cette opération permet de coudre une face B sur les arêtes A_E si elles sont libres. Nous avons la possibilité avec cette règle d'ajouter des faces dans notre modèle et donc de faire croître la feuille. C'est avec cette règle que nous pourrons faire apparaître les nervures primaires puis les nervures secondaires d'une feuille.

La grammaire 2.1 nous montre la mise en pratique de cette règle de croissance. Deux faces, A et B , sont tout d'abord définies dans le préambule. Ensuite l'axiome est donné comme étant une face A et finalement la règle $p00$ ajoute une face B sur les arêtes support E des faces A .

Grammaire 2.1. @ Définition des faces @

#define A(4, 1, 0, 0, 0, 1, 1)

#define B(4, 1, 0, 0, 0, 1, 1)

@ Axiome @

#axiome : A

@ Règles @

 $p00 A \rightarrow A[B]_E$

L'ajout d'une face est toujours effectué par rapport à son arête origine. Si l'arête support spécifiée est, par exemple, l'arête extrémité comme dans la règle $p00$ de la grammaire 2.1. L'arête origine d'une face B sera cousue sur les arêtes extrémités libres des faces A existantes. Nous pouvons voir le résultat de la grammaire 2.1 sur la Figure 2.2.

La grammaire 2.1 peut facilement être modifiée avec une autre règle de croissance pour simuler le développement d'une nervure. Il suffit de rajouter le collage d'une face A sur les faces B pour avoir un allongement du modèle. La nouvelle grammaire 2.2 inclue cette modification et fait croître une nervure.

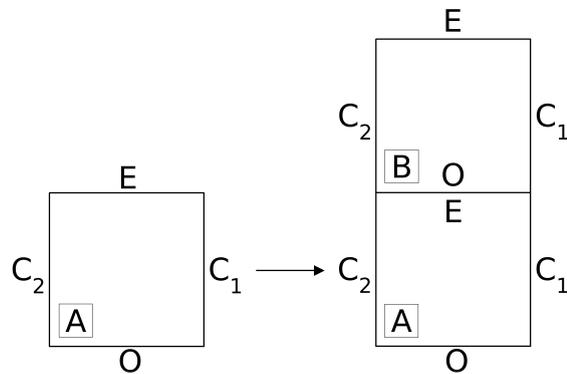


FIGURE 2.2 – Une face B est collée par son arête origine sur les arêtes extrémité libres des faces A .

Grammaire 2.2.

@ Définition des faces @
#define A(4, 1, 0, 0, 0, 1, 1)
#define B(4, 1, 0, 0, 0, 1, 1)

@ Axiome @
#axiome : A

@ Règles @
p00 A → A[B]_E
p01 B → B[A]_E

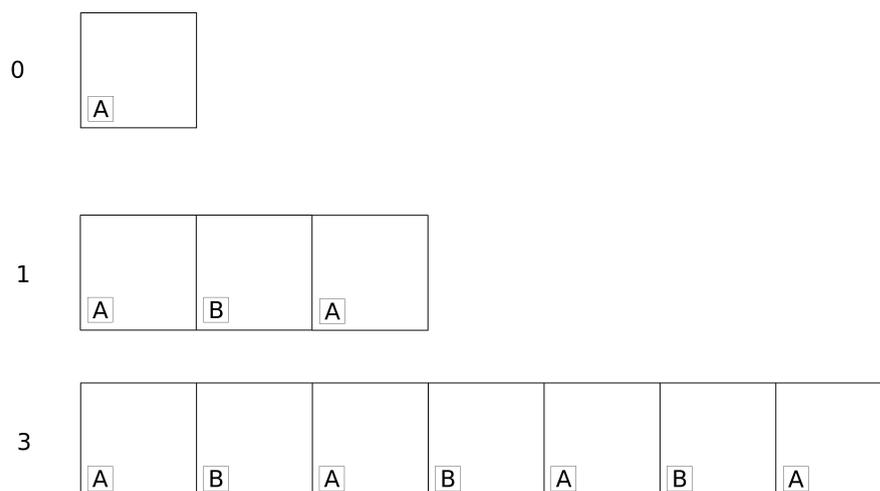


FIGURE 2.3 – La croissance du modèle à partir d'une face A . nous pouvons observer l'état initial puis les itérations 1 et 3.

La croissance induite par la grammaire 2.2 peut être observée sur la Figure 2.3 à l'état initial pour 1 itération puis pour 5 itérations. Nous pouvons de plus voir à la Figure 2.4 le résultat de cette grammaire.

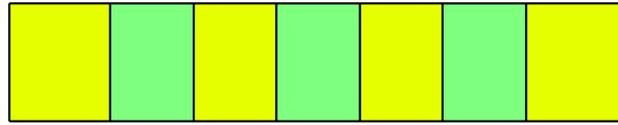


FIGURE 2.4 – L’itération numéro 3 de la grammaire 2.2 visualisée grâce à notre modèle. Les faces jaunes ont pour label A , les vertes B .

Collage :

Le second type de règle utilisé est le collage. Les règles de collage permettent de coudre une arête ayant un label donné avec une autre arête. Ces deux arêtes, si elles existent et si elles possèdent une adjacence au niveau des brins, seront cousues entre elles. De plus, ce type de règle nous permet d’introduire une dépendance au contexte dans notre modèle. Voici le prototype d’une règle de collage :

$$A : A_{C_1} < B_{C_2} \rightarrow A_{C_1} | B_{C_2}$$

Ici, si une arête A_{C_1} est adjacente à une arête B_{C_2} , alors les deux arêtes sont cousues entre elles. Nous pouvons observer l’illustration de cette règle sur la figure 2.5.

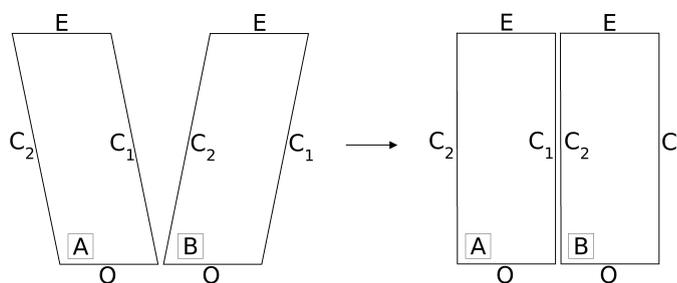


FIGURE 2.5 – Le collage de deux arêtes adjacentes. Ici quand on trouve une arête A_{C_1} adjacente à une arête B_{C_2} , ces deux arêtes sont collées.

Cette règle permet de combler les espaces de notre modèle. Elle peut par exemple coller deux nervures après leurs croissances en parallèle. La grammaire 2.3 nous montre l’utilisation d’une telle règle.

Grammaire 2.3. @ Définition des faces @
 #define A(4, 1, 0, 0, 0, 1, 1)
 #define B(4, 1, 0, 0, 0, 1, 1)

@ Axiome @
 #axiome : A

@ Règles de Croissance@
 p00 $B \rightarrow B[B]_E$

p01 $A \rightarrow A[B]_{C_1}$

p02 $A \rightarrow A[B]_E$

@ Règles de Collage@
 p03 $B : B_{C_1} < B_{C_2} \rightarrow B_{C_1}|B_{C_2}$

Avec la grammaire 2.3 des faces B sont ajoutées sur les arêtes A_{C_1} et A_E (règles $p01$ et $p02$). Ensuite, nous ajoutons une phase de croissance grâce à la règle $p00$ qui nous permet de faire croître deux nervures à partir des faces B . Finalement, les deux nervures sont collées entre elles en utilisant la règle $p03$ qui spécifie que chaque arête B_{C_1} et B_{C_2} adjacentes seront collées. Le résultat de la grammaire 2.3 peut être observé sur la Figure 2.6.

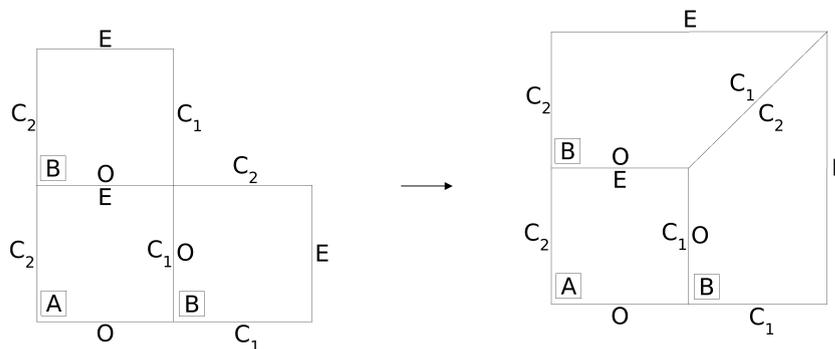


FIGURE 2.6 – Le résultat de la première itération de la grammaire 2.3. Deux faces B sont ajoutés sur les arêtes A_{C_1} et A_E puis les arêtes B_{C_1} et B_{C_2} sont collées.

Eclatement de face :

Le dernier type de règle de base est l'éclatement de face. Ces règles sont utilisées pour diviser une face en deux autres faces parallèlement à une arête donnée. Une règle de découpe sera définie de cette façon :

$$A \rightarrow \circ BC$$

Dans cet exemple, les faces A sont découpées en deux faces B et C en prenant comme support les arêtes A_O . Une telle règle peut être vue en action dans la grammaire 2.4.

Grammaire 2.4. @ Définition des faces @

```
#define A(4, 1, 0, 0, 0, 2, 1)
```

```
#define B(4, 1, 0, 0, 0, 1, 1)
```

```
#define C(4, 1, 0, 0, 0, 1, 1)
```

@ Axiome @

```
#axiome : A
```

@ Règles de Découpe@

```
p00A → oBC
```

Ces règles nous offrent la possibilité d'effectuer des découpes sur toute la longueur du modèle suivant une arête support définie. Nous pouvons par exemple diviser une nervure par son centre ou bien faire apparaître un pavage régulier à partir d'une face.

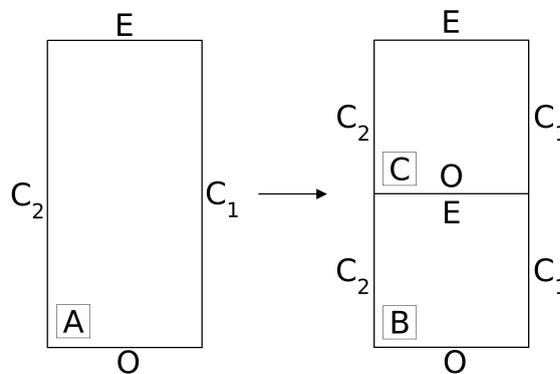


FIGURE 2.7 – La découpe d'une face A en deux faces B et C suivant l'arête A_O . Cette figure nous montre le résultat de la grammaire 2.4.

Grâce à ces trois types de règles, nous pouvons développer les nervures principales d'une feuille. En effet, la croissance et le collage nous permettent de créer la forme principale puis la découpe donne la possibilité de raffiner les faces créées. Cependant, nous avons voulu enrichir ce système de grammaire. Le modèle des 2-G-Cartes L-Systèmes décrit précédemment se voit doté de plusieurs améliorations visant à rendre plus intuitive et performante la création de grammaire.

2.3.2.2 Amélioration de la grammaire

Pour rendre la création de feuilles plus facile, nous avons inclus différents outils dans le modèle des 2-G-Cartes L-Systèmes. Ces outils permettront d'ajouter un

aspect paramétrable pour rendre notre modèle beaucoup plus fluide et intuitif à manipuler.

Trois types d'améliorations ont été mise en oeuvre dans notre modèle. Le premier vise à donner la possibilité à l'utilisateur d'utiliser des variables. Le second permet d'introduire un langage de script dans les règles et donc, de combiner les variables ou bien de faire appel à des fonctions mathématiques. Enfin des améliorations sont apportées à la définition des règles en permettant de surcharger les attributs d'une face, d'inclure un aspect conditionnel ou bien de réutiliser les données d'une face existante comme dans les L-Systèmes classiques.

Variables :

Des variables peuvent être définies dans la grammaire. Leur identifiant ainsi que leur valeur doivent être donnés à la suite de la définition des faces. Nous pouvons voir la définition de variables dans la grammaire 2.5 avec l'exemple de la variable *longueur*.

Grammaire 2.5. @ Définition des faces @
`#define A(4, 1, 0, 0, 0, 1, 1)`
`#define B(4, 1, 0, 0, 0, 1, 1)`

@ Variables @
`#define longueur = 2`

@ Axiome @
`#axiome : A`

@ Règles@
`p00 A → A[B]E`
`p00 B → B[B]E`

Cette variable n'est pour le moment pas utilisée dans la liste des règles. Cependant, nous pouvons l'utiliser dans une définition de variable suivante. Une telle modification est apportée à la grammaire 2.6. Dans cette grammaire la variable *largeur* vaudra le double de la variable *longueur*.

Grammaire 2.6. ...
 @ Variables @
`#define longueur = 2`
`#define largeur = 2 * longueur`
 ...

Enfin, deux variables sont implicitement définies dans chaque grammaire. La première est la variable *etape* qui donne le numéro de l'itération courante durant

le processus de création. La seconde est la variable *etapeFinale* qui donnera le numéro de la dernière itération. Ces deux variables spéciales pourront être utilisées par la suite dans les blocs de script et les règles.

La première servira à changer le comportement de la grammaire lorsqu'un certain nombre d'itérations sera passé. Nous pourrons par exemple faire se diviser les nervures lorsque nous arrivons au milieu du processus de création ou bien faire augmenter la longueur des faces collées en fonction du numéro de l'itération courante.

La seconde variable permettra quant à elle de savoir le numéro de la dernière itération. Cela nous sera utile pour permettre une phase de finalisation dans nos grammaires. En effet, si nous nous trouvons à la dernière itération, nous pourrons changer le comportement de la grammaire. La dernière itération permettra de gérer, par exemple, la création de bords dentelés pour une feuille.

Les variables seront utilisées dans des blocs particuliers appelés blocs de script. Ces blocs sont délimités par "<" et ">" et incluent du code Lua. Ce code est un langage de script qui nous permettra d'utiliser des fonctions mathématiques et de combiner nos variables. Le langage Lua est portable et robuste, il possède de plus une syntaxe proche du langage C.

Surcharge :

La surcharge est un outil mis en place pour pouvoir changer la valeur de n'importe quel attribut d'une face. En effet, pour avoir une grammaire encore plus paramétrable, il nous faut la possibilité de changer les valeurs données aux faces en fonction des éléments de la génération. Ces différents facteurs peuvent être le numéro de l'itération courante, la valeur d'une variable ou bien la proximité de l'étape finale.

Pour ce faire, nous ajoutons la surcharge des paramètres lors de la définition d'une règle de croissance. Une parenthèse est ajoutée en redéfinissant ou non les attributs de la face concernée. Le prototype de la surcharge de paramètres est la suivante :

$$A \rightarrow A[B(n, t, \alpha, \beta, \gamma, l, w)]_E$$

Nous pouvons voir que chacun des paramètres de base constituant la définition d'une face peut être surchargé. Pour l'utilisation effective de la surcharge, nous laissons vides les paramètres que nous ne voulons pas surcharger. Ils garderont les valeurs par défaut données lors de la définition de la face. Un tel ajout peut être observé dans la grammaire 2.7 avec la surcharge de la longueur des faces A ajoutées ainsi que de leur coefficient de translation.

Grammaire 2.7. @ Définition des faces @
 #define A(4, 1, 0, 0, 0, 1, 1)

 @ Axiome @
 #axiome : A

 @ Règles @
 p00 A → A[A(, 2, , , , 2,)]_E



FIGURE 2.8 – Surcharge des faces A . Nous pouvons voir ici le résultat de la grammaire 2.7 pour 6 itérations.

Grammaire 2.8. @ Définition des faces @
 #define A(4, 10, 0, 0, 0, 10, 1)
 #define B(4, 5, 0, 0, 0, 5, 1)
 #define C(4, 5, 0, 0, 0, 5, 1)

 @ Axiome @
 #axiome : A
 #define angle = α

 @ Règles @
 p00 A → A[B]_E
 p01 B → B[A]_E

 p02 B → B[C(, , , < angle > , ,)]_{C1}
 p03 B → B[C(, , , < -angle > , ,)]_{C2}

 p04 C → C[C]_E

La grammaire 2.7 s'initialisera avec l'axiome A ayant pour paramètres les données de base de la définition. Cependant, lors du premier ajout, les faces A ajoutées auront une longueur de deux unités. Nous pouvons observer la surcharge des faces A sur la Figure 2.8. l'axiome (sur la gauche) est deux fois moins long que les faces A ajoutées par la suite.

Tous les paramètres peuvent être surchargés. Cet outil est très utile pour redéfinir les angles associés aux faces et ainsi créer des nervures secondaires plus ou moins

espacées. Nous pouvons voir cette mise en oeuvre dans la grammaire 2.8. Une variable “*angle*” est définie et réutilisée dans la surcharge pour ajouter les faces *C* de part et d’autre des faces *B*. Les résultats obtenus peuvent être observés sur la Figure 2.9.

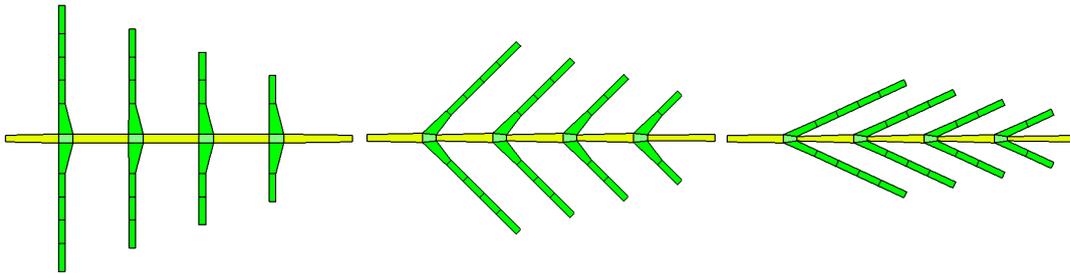


FIGURE 2.9 – Les résultats de la grammaire 2.8 pour des angles α valant 90, 45 et 25 degrés de gauche à droite.

L’angle peut aussi être changé lors de la croissance complète des nervures. Plutôt que de fixer l’angle de départ des nervures, un tel traitement permettra d’incurver une nervure sur toute sa longueur et sur n’importe quel axe. De ce fait, le réseau de nervures constitué de subdivisions en deux dimensions de l’espace est plongé en trois dimensions. Nous pouvons voir un tel réseau de nervures sur la Figure 2.10.

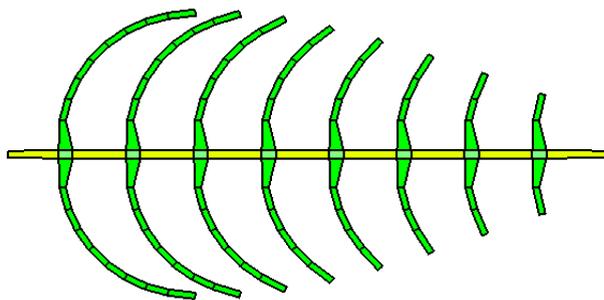


FIGURE 2.10 – La création de nervures secondaires courbées grâce à la surcharge des paramètres. Chaque face est ajoutée lors de la croissance des nervures secondaires avec un angle différent de son angle d’origine.

Conditions :

Pour changer le comportement de la grammaire au cours de la génération, nous avons introduit des conditions dans les règles. L’utilisateur aura la possibilité de définir des blocs spéciaux permettant d’exécuter une règle sous certaines conditions. Le prototype d’une règle obéissant à une condition est le suivant :

$$A \{Bloc1\} \{Bloc2\} \{Bloc3\} \rightarrow A[B]_E$$

Le premier bloc nommé (*Bloc1*) sera interprété dans tous les cas. Les opérations contenues dans ce bloc représentent la phase d'initialisation de la règle. Le second bloc (*Bloc2*) définira quant à lui la condition. Cette condition pourra porter sur des variables définies au début de la grammaire ou bien sur des variables globales telles que l'itération courante ou l'itération finale. Le dernier bloc (*Bloc3*) sera interprété seulement si la condition donnée dans le second bloc est vérifiée. Finalement, le corps de la règle sera lui aussi interprété uniquement en cas de succès de la condition. La grammaire 2.9 illustre l'utilisation de cette règle conditionnelle.

Grammaire 2.9.

```
@ Définition des faces @
#define A(4, 10, 0, 0, 0, 10, 1)
#define B(4, 5, 0, 0, 0, 5, 1)
#define C(3, 5, 0, 0, 0, 5, 1)

@ Axiome @
#axiome : A
#define angle = 20

@ Règles @
p00 A → A[B]E
p01 B → B[A]E

p02 A{ }{ etape == etapeFinale/2 }{ } → A[C]E

p03 C → C[A(, , , < angle > , , )]E
p04 C → C[A(, , , < -angle > , , )]C1
```

Dans la grammaire 2.9 nous utilisons trois faces différentes. La face *A* va représenter la face de croissance. La face *B* permettra de créer les noeuds des nervures secondaires. Enfin, la face *C*, ne comportant que trois arêtes, servira pour diviser la nervure principale en deux.

Les règles *p00* et *p01* assurent la croissance de la nervure principale en alternant les faces *A* et *B*. Ensuite, la règle *p02* ajoute une face triangulaire *C* à la fin de l'arête principale si et seulement si l'étape courante est égale à l'étape finale. En d'autres termes, si la génération de la feuille est à mi-chemin. Finalement, les règles *p03* et *p04* permettent d'ajouter deux faces *A* de part et d'autre de la face *C* triangulaire.

La génération se poursuit ensuite non plus avec une seule, mais avec deux nervures principales. Le résultat de cette génération est visible à la Figure 2.11. Les règles conditionnelles nous permettent d'inclure dans notre modèle la possibilité

de générer des formes très complexes.

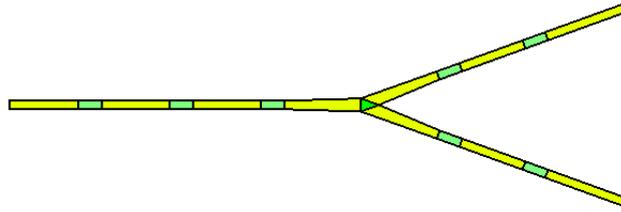


FIGURE 2.11 – Ici la nervure principale se divise en deux à mi-génération grâce à une règle conditionnelle.

Réutilisation :

Le dernier outil inclus au sein des règles est la possibilité de réutiliser les paramètres de la face mère. Cette amélioration donne à l'utilisateur la possibilité de propager une information tout au long de la génération des nervures. En effet, lors de la croissance d'une nervure, la grammaire pourra se baser sur la taille des faces mères pour définir celles des faces ajoutées au modèle. Cette réutilisation dépend de la syntaxe suivante :

$$A(a, b, c, d, e, f, g) \rightarrow A[B(a, b, , , ,)]_E$$

Chaque paramètre de la face mère se voit donner un nom qui n'aura de visibilité que pour la règle en question. Utilisée seule, cette amélioration n'apporte rien. Cependant, avec la surcharge de paramètres et les blocs conditionnels, les paramètres récupérés de la face mère pourront être réutilisés. Ici nous ne réutilisons que les paramètres a et b . Un tel traitement nous permet de transmettre les informations des faces mères aux faces ajoutées lors de la croissance.

Autres outils :

Une fois le réseau de nervures défini par les règles de la grammaire, il nous reste à générer le limbe de la feuille. Le limbe est le corps proprement dit de cette feuille et regroupe les cellules permettant au végétal de respirer et réaliser sa photosynthèse. Ce limbe va englober tout le réseau de nervures et en dépendra entièrement.

De ce fait, nous avons opté pour une génération automatique pendant la construction du réseau de nervures. A chaque étape, nous remplissons les nervures générées avec des faces représentant le limbe de la feuille. Cette phase de remplissage nécessite des informations relatives à l'agencement des nervures de la feuille.

Les extrémités : ce sont les arêtes extrémités E appartenant aux nervures qui sont libres par α_2 . Ces arêtes constituent les jalons que nous utiliserons pour créer le limbe de la feuille.

Le bord des nervures : ce bord est représenté par toutes les arêtes libres par α_2 entre deux extrémités.

L'ordre des axes : C'est un indice donnant l'ordre de chaque nervure. Une nervure peut être primaire (la base de la feuille), secondaire (partant des nervures primaires), ou bien d'ordre supérieur. Cet ordre est calculé automatiquement en fonction de l'emplacement du collage de la face. Si elle est collée sur un côté de la face mère, l'ordre sera incrémenté par rapport à celui de la face mère, sinon il restera inchangé. Il nous permettra de savoir si du limbe doit être créé ou si l'espace entre les nervures doit rester vide.

Le principe du remplissage des éléments internes de la feuille repose sur le parcours des bords de la carte d'une extrémité à l'autre tout en créant les faces de remplissage adéquates. Lors de ce parcours, les brins ainsi identifiés sont sauvegardés pour servir de base à la nouvelle face. Les nouveaux brins sont ensuite créés sur le modèle des brins sauvegardés et la face est terminée au moyen d'une nouvelle arête entre le premier brin et le dernier brin de la face. Enfin, les coutures topologiques entre ces brins sont créées et la face ajoutée puis cousue au reste de la structure.

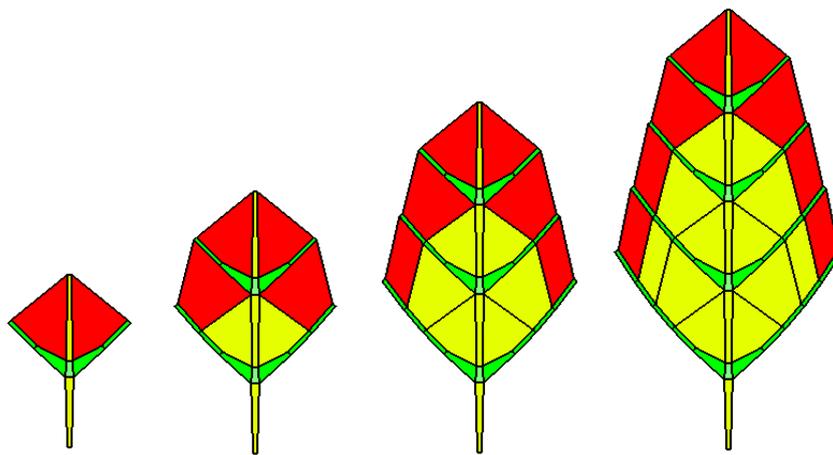


FIGURE 2.12 – Le remplissage au cours des itérations. Les faces rouges sont les faces du limbe ajoutées à chaque itération à la suite la génération des nervures.

Cette phase de remplissage se déroule à chaque itération. En effet, ce traitement est nécessaire à chaque pas de la génération pour permettre au limbe d'épouser la forme du réseau de nervures. La Figure 2.12 montre une telle génération automatique. De gauche à droite nous pouvons voir de la première à la quatrième itération de la génération d'une feuille. Nous pouvons observer qu'aucune face de remplissage n'est ajoutée entre la tige de la feuille et les premières nervures secondaires.

Nous avons de plus inclus dans la grammaire une variable implicite *remplissage* pouvant gérer la création des faces représentant le limbe à chaque étape. Si cette variable n'est pas spécifiée, le remplissage des éléments internes se fait de la première à la dernière itération. L'utilisateur peut aussi donner une valeur à cette variable pendant la déclaration des variables. Si la valeur donnée est -1 alors le remplissage aura lieu comme si de rien n'était, une autre valeur définira l'arrêt du remplissage.

Par exemple sur la Figure 2.13 La variable de remplissage est fixée à zéro pour la génération de gauche : on ne voit que les nervures de la feuille. Au centre cette variable vaut cinq ce qui induit un remplissage du limbe jusqu'au milieu de la génération. Enfin, la feuille de gauche est totalement remplie par le limbe en fixant la variable à dix c'est-à-dire l'itération finale.

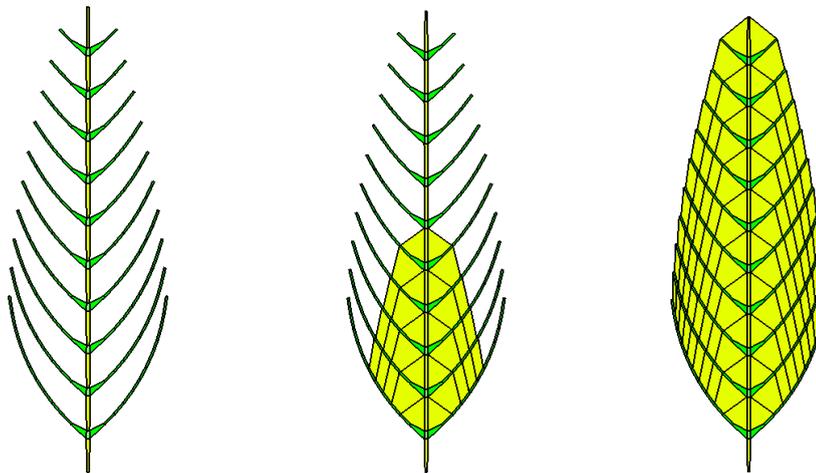


FIGURE 2.13 – De gauche à droite : la variable est fixée à zéro, pour ne garder que le réseau de nervures puis à cinq pour ne garder que la moitié du limbe et enfin à dix pour avoir tout le corps de la feuille.

2.3.2.3 Création de grande quantité de feuilles

Grâce aux différentes règles et améliorations de la grammaire, nous pouvons construire la structure utilisée pour simuler les nervures d'une feuille. Une espèce de feuille donnée possède un réseau de nervures qui lui est propre. Ce réseau peut varier légèrement d'une feuille à l'autre au sein de la même espèce, mais il ne se produira que des variations minimales.

Deux feuilles pourront avoir un petit écart dans leurs longueurs ou bien dans leurs largeurs. Les nervures d'une feuille pourront être un peu moins inclinées qu'une autre. Enfin, les courbures de leurs nervures principales pourront différer de quelques degrés. Cependant, toutes les feuilles d'une espèce donnée obéiront aux mêmes règles concernant la croissance de leurs nervures : elles auront une structure et donc une topologie identique même si leur géométrie varie.

En utilisant le système de scripts et des valeurs aléatoires, nous pouvons générer une feuille différente pour chaque génération. Les valeurs aléatoires seront fixées dans une fourchette correspondant à l'espèce de feuille désirée. Par exemple, des nervures avec un angle de 45 degrés seront définies entre 40 et 50 degrés pour respecter la forme de l'espèce.

Nous pouvons voir sur la grammaire 2.10 la définition d'une feuille très simple. Elle possède une nervure principale ainsi que des nervures secondaires collées suivant un angle "*angleSecondaire*" et courbées suivant un angle "*angleCourbure*". Cette grammaire permet de générer la feuille vue à la Figure 2.14.

Grammaire 2.10.

```
@ Définition des faces @
#define A(4, 20, 0, 0, 0, 20, 1)
#define B(4, 10, 0, 0, 0, 10, 1)
#define C(4, 10, 0, 0, 0, 10, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
@ Axiome @
#axiome : A
#define angleSecondaire = 50
#define angleCourbure = 4
@ Règles @
p00 A → A[B]E
p01 B → B[A]E
p02 B → B[C(, , , , < angleSecondaire > , , )]C1
p03 B → B[D(, , , , < -angleSecondaire > , , )]C2
p04 C → C[C(, , , , < -angleCourbure > , , )]E
p05 D → D[D(, , , , < angleCourbure > , , )]E
```

Nous allons rajouter des valeurs aléatoires dans la génération de cette feuille. L'angle *angleSecondaire* sera définie entre 45 et 55 degrés. De même, l'angle *angleCourbure* se trouvera entre 2 et 6 degrés. Nous pouvons voir les changements sur la Figure 2.14, cette feuille n'est plus du tout symétrique et régulière. De plus, chaque génération nous donne une feuille différente.

Nous avons de plus la possibilité d'agir aussi sur tous les paramètres des faces composant une feuille. Nous pouvons par exemple faire varier les dimensions des éléments composant les nervures pour changer la taille globale de la feuille. La grammaire 2.11 comprend l'ajout de paramètres aléatoires sur la taille des éléments de la nervure principale ainsi que sur les angles d'accroche et de courbure des nervures secondaires. Ces modifications sont visibles sur la Figure 2.15.

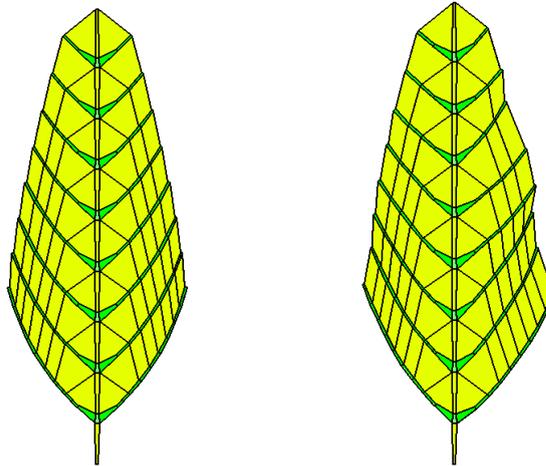


FIGURE 2.14 – Gauche : le résultat de la génération de la grammaire 2.10 pour 8 itérations. Droite : l'introduction d'angles aléatoires dans la grammaire 2.10. La feuille parait plus réaliste, car moins "parfaite".

Grammaire 2.11. @ Définition des faces @

```
#define A(4, 20, 0, 0, 0, 20, 1)
#define B(4, 10, 0, 0, 0, 10, 1)
#define C(4, 10, 0, 0, 0, 10, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
```

@ Axiome @

```
#axiome : A
```

@ Règles @

```
p00 A → A[B]E
p01 B{l = alea(15, 25)}{1}{1} → B[A(< l >, ,, 0, < l >, )]E
p02 B → B[C(,,, < alea(50, 60) >, ,)]C1
p03 B → B[D(,,, < -alea(50, 60) >, ,)]C2
p04 C → C[C(,,, < -alea(2, 6) >, ,)]E
p05 D → D[D(,,, < alea(2, 6) >, ,)]E
```

Nous pouvons voir sur la Figure 2.16 la variation des angles et longueurs des faces d'une feuille de tilleul. Ces feuilles appartiennent toutes à la même espèce, mais sont quand même toutes différentes. Ici la génération de la collection de feuilles tilleul est faite sans faire appel à des paramètres aléatoires.

La Figure 2.17 montre le résultat de 20 générations successives à partir d'une seule grammaire. Cette grammaire contient des angles définis dans une fourchette plutôt que fixés avec une unique valeur. Leurs valeurs sont ensuite déterminées aléatoirement dans la fourchette donnée lors de l'ajout d'une face. Nous pouvons voir que chaque feuille est différente des autres feuilles de la génération.

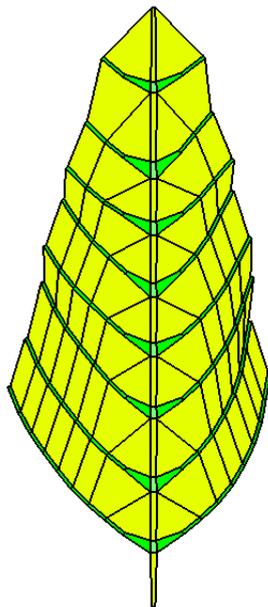


FIGURE 2.15 – Ici, un des résultats donnés par la grammaire 2.11 avec l'ajout de paramètres aléatoires. Nous pouvons voir les tailles différentes pour les faces de la nervure principale.

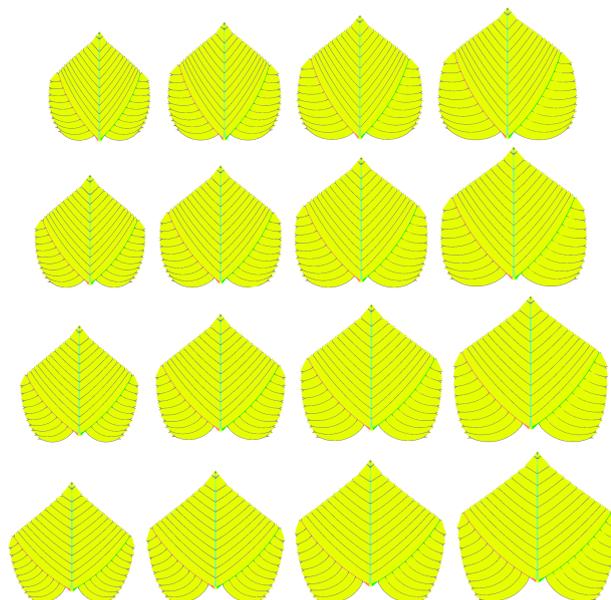


FIGURE 2.16 – Les différentes feuilles de tilleul générées. La longueur des nervures augmente de gauche à droite et les valeurs des angles de haut en bas.

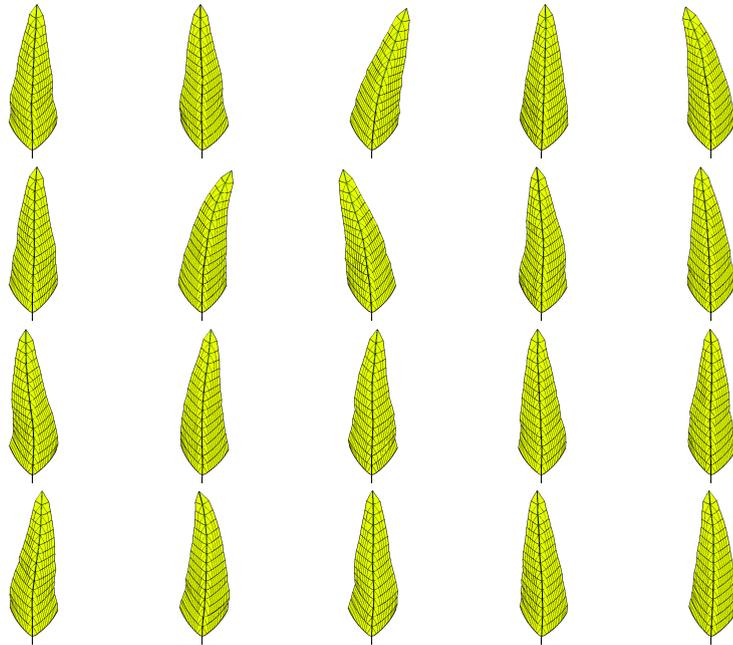


FIGURE 2.17 – Ces 20 feuilles sont générées à partir de la même grammaire. Les différences inter-espèce sont dues à l’ajout de variables aléatoires pour les angles des nervures.

Enfin grâce aux différentes itérations de la grammaire, nous pouvons gérer des niveaux de détail. En effet, la grammaire génère les nervures principales de la feuille. Ces nervures donnent la forme générale de la feuille depuis les premières jusqu’aux dernières itérations. En restant dans une fourchette d’itérations relativement confortable, nous pouvons générer des feuilles avec plus ou moins de détails. Nous pouvons voir l’application de ces niveaux de détail sur la Figure 2.18

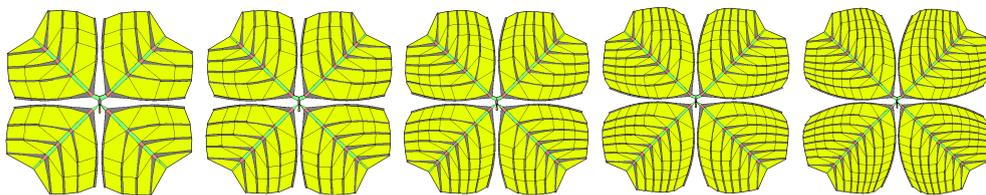


FIGURE 2.18 – Nous pouvons observer sur cette figure cinq niveaux de détail pour une feuille de trèfle. De gauche à droite nous voyons les itérations 5, 6, 7 et 8 de la grammaire pour créer ce trèfle à quatre feuilles. Le nombre de faces utilisées augmente de gauche à droite en suivant le nombre d’itérations.

Toutes ces règles et améliorations nous permettent de générer des feuilles complexes composées de subdivisions de l’espace en deux dimensions plongées en trois dimensions. Nous avons maintenant la possibilité de générer un nombre quelconque de feuilles, appartenant toutes à la même espèce, en utilisant une seule et unique grammaire. Cette grammaire peut être paramétrée de telle sorte que le réseau de nervure s’ajuste au mieux au modèle de la feuille réelle. La prochaine étape est de pouvoir donner aux feuilles générées un âge donné. Ce vieillissement

devra se faire aussi bien au niveau géométrique que colorimétrique.

2.4 Vieillessement de la feuille

Durant toute leur vie, les feuilles changent beaucoup d'aspect. Tout d'abord petites, tendres et claires à la sortie du bourgeon au printemps, elles atteignent leur taille adulte au coeur de l'été. Cette phase permet au végétal de réaliser la photosynthèse grâce à la chlorophylle qui donne aux feuilles leurs couleurs vertes. Enfin pendant l'automne, l'illumination se faisant moins importante, les feuilles deviennent inutiles. C'est à cette époque qu'elles commencent à perdre leur chlorophylle et deviennent plus sombres et rouges. Dans un même temps, leurs formes changent en se flétrissant et se recroquevillant. Enfin, les feuilles tombent au pied de l'arbre pour se décomposer et être assimilées.

Toutes ces phases de vieillissement comportent des aspects bien distincts. Premièrement, la forme de la feuille ne commence à être affectée que lorsque celle-ci va mourir. Ensuite sa couleur varie tout au long de sa vie en fonction du taux de chlorophylle présent dans la feuille. Enfin des attaques extérieures peuvent causer des dommages aux feuilles, et ce, à n'importe quelle phase de son vieillissement.

2.4.1 Aspects géométriques du vieillissement

Le vieillissement géométrique sera utilisé pour simuler le recroquevillement des feuilles lors de leurs séchages. Il pourra aussi être utilisé pour simuler un défaut dans une feuille comme, par exemple, une croissance gênée par un obstacle. L'aspect géométrique du vieillissement sera intégralement géré par la grammaire en s'aidant des améliorations qui lui ont été apportées.

Nous proposons de caractériser l'âge d'une feuille avec des éléments propres à la grammaire :

Le nombre de nervures d'une feuille. Ce point particulier du vieillissement est une approximation du processus biologique de développement des feuilles. En effet, les principales nervures de la feuille sont déjà créées dans le bourgeon et ce n'est que par la suite que les nervures secondaires apparaissent dans le limbe de la feuille. Toutefois, notre grammaire permet de contrôler la forme d'une feuille avec seulement quelques nervures appropriées. Ce premier paramètre est directement contrôlé par le nombre d'itérations assigné à la génération. Nous pouvons voir sur la Figure 2.19 une telle croissance.

La longueur des nervures. Le processus de croissance d'une feuille fait grandir les nervures en longueur jusqu'à leurs tailles adultes. Ce processus commence dès

la sortie du bourgeon affecte la taille de toutes les nervures. Ce paramètre de la croissance des feuilles est géré par à la fois le nombre d'itérations qui rajoute des éléments aux nervures et par la taille des faces composant les nervures.

La figure 2.19 illustre l'accroissement de la taille des nervures lors de la croissance d'une feuille ainsi que trois étapes de la croissance d'une feuille de châtaignier. La première étape sur la gauche illustre la feuille après la sortie du bourgeon. Les deux étapes suivantes montrent la croissance de la feuille en suivant notre approximation du processus biologique de croissance d'une feuille.

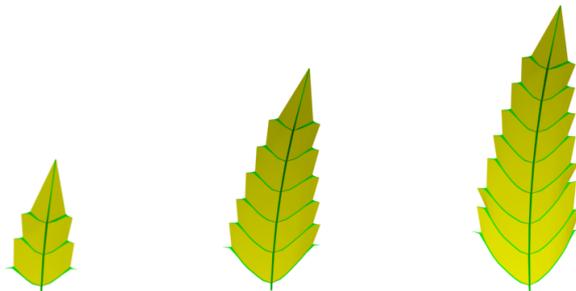


FIGURE 2.19 – De gauche à droite nous pouvons voir la croissance d'une feuille de châtaignier de la sortie du bourgeon à sa taille adulte.

Les rotations de chaque nervure. Une fois qu'une feuille a atteint son âge adulte, elle commence à vieillir et à se détériorer. Ce cycle de vieillissement induit un processus de recroquevillement de la feuille. En effet, après un certain temps qui dépend de facteurs extérieurs tels que l'humidité, l'ensoleillement ou bien les contacts avec le sol, une feuille va complètement s'enrouler sur elle-même en suivant l'axe de ses nervures. Ces nervures ne sont pas faites du même matériau que le limbe, c'est pour cela que ces changements géométriques suivent la structure des nervures. Nous pouvons complètement simuler ce processus grâce aux angles d'orientation α , β et γ inclus dans notre modèle. Un tel changement est visible sur la Figure 2.20.

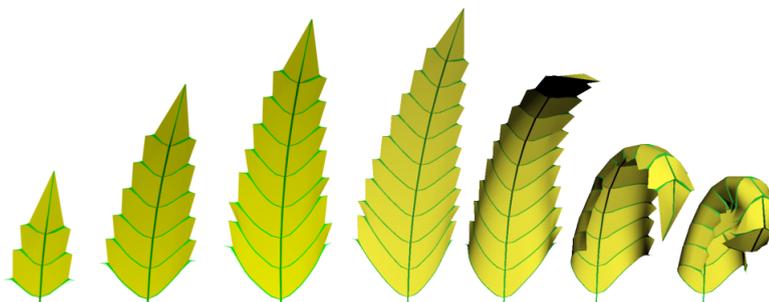


FIGURE 2.20 – Le cycle complet du vieillissement géométrique d'une feuille de châtaignier. De la gauche pour la naissance à la droite pour la fin de vie, nous pouvons observer les différentes étapes du vieillissement de cette feuille.

Un dernier aspect géométrique du vieillissement fait référence aux dommages

subis par une feuille. Tout au long de sa vie, une feuille peut subir différentes attaques dues à des éléments extérieurs tels que des animaux, des insectes, des maladies, des parasites ou bien des intempéries. Les modifications importantes telles qu'une feuille découpée par la mâchoire d'un animal ou bien déchirée par le passage trop proche d'un élément extérieur sont directement applicables grâce aux règles et améliorations de la grammaire. La Figure 2.21 nous montre les dégradations pouvant être incluses dans la grammaire.

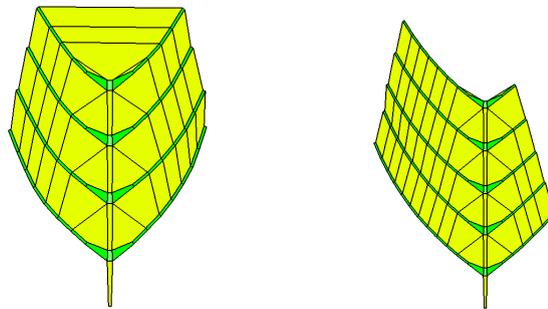


FIGURE 2.21 – Deux dégradations possible dues à des éléments externes. Ici la pointe de la feuille de gauche s'est vue amputée, la feuille de droite a subi une déchirure.

Les attaques d'insectes sont aussi incluses dans notre modèle. Les insectes et plus particulièrement les chenilles réalisent de grosses dégradations sur les feuilles. Ces attaques sont principalement concentrées sur le limbe de la feuille et non sur les nervures. En effet, le limbe possède beaucoup plus de substances nutritives que les nervures. De plus, la taille de ces insectes varie considérablement, ce qui induit des dégradations plus ou moins grosses et localisées suivant le type d'insecte.

Nous avons développé une génération automatique de trous faits par des insectes. Ces trous sont placés empiriquement tout en connaissant les éléments constituant la feuille ainsi que la position des nervures. Pour réaliser de telles attaques, nous générons une carte d'opacité permettant de savoir où sont les trous dans le limbe de la feuille. Nous utilisons une distribution aléatoire des chenilles sur la feuille, elles dégradent ensuite la feuille suivant trois critères :

- Le rayon des trous effectués par la chenille. Ce rayon est significatif de la taille de la chenille.
- La direction de déplacement de la chenille. Elle peut se déplacer en ligne droite ou bien suivre les nervures du végétal.
- Le nombre d'attaques subit par la feuille. Cela nous permet de savoir combien de fois la chenille s'en prendra à la feuille.



FIGURE 2.22 – Sur la gauche, une attaque de chenille sur une feuille réelle. Sur la droite nous pouvons voir la génération des attaques d’insectes, les parties rouges sont les morceaux de feuille manquants.

Nous pouvons observer sur la Figure 2.22 l’image d’une vraie feuille attaquée par des insectes. Les trous effectués par les insectes évitent les nervures et ont des tailles différentes. Sur la droite de cette Figure, nous voyons le résultat de la génération d’attaques d’insectes sur une feuille. Les nervures sont représentées en noir et les attaques en rouges. Les zones rouges montrent où de la matière a été enlevée. Nous pouvons voir que ces trous évitent les nervures et se concentrent sur le limbe. De plus, plusieurs tailles d’insectes ont été simulées ce qui nous donne des tailles de trous différentes.

2.4.2 Vieillessement colorimétrique

L’autre aspect du vieillissement d’une feuille est la modification de ses couleurs. En effet, les feuilles sont classiquement représentées avec une couleur verte, cependant, certaines peuvent être d’une couleur différente dès leur croissance. De plus, la majorité des feuilles changent de couleur en automne et deviennent marron et rouge. Ces changements sont causés par la décomposition de la chlorophylle, substance permettant à la feuille de réaliser la photosynthèse, qui donne une couleur verte à la majorité des feuilles.

En se décomposant, la chlorophylle laisse apparaître les deux autres pigments présents dans les feuilles, à savoir les anthocyanes et les caroténoïdes qui donnent respectivement une couleur rouge et jaune-orange aux feuilles. Ces deux pigments ont le rôle de protéger la feuille des températures plus faibles lorsque la chlorophylle est décomposée. Cette chlorophylle disparaît plus rapidement près des nervures à cause du flux de sève dans la feuille alors qu’elle disparaît moins vite au centre du limbe. Néanmoins, ce phénomène de dégradation peut survenir à des endroits aléatoires sur une feuille en fonction de divers paramètres comme l’ensoleillement ou les dommages subits par la feuille.

Notre méthode permet de générer la couleur d’une feuille à n’importe quel âge. Elle est basée sur deux paramètres donnés par l’utilisateur ainsi :

- Deux couleurs fournies par l'utilisateur. Ces deux couleurs définiront l'état "jeune" et l'état "vieux" d'une feuille. Spécifier une couleur jeune et une couleur vieille nous permet de générer la texture d'une feuille à n'importe quel âge. De plus, l'utilisateur peut définir ces deux couleurs en fonction de l'espèce de feuille qu'il désire générer.
- Un paramètre de mélange. Il est utilisé pour déterminer la quantité de chaque couleur fournie (jeune et vieille) utilisée pour générer la texture finale. Ce paramètre est directement en relation avec l'âge de la feuille, il permet de générer une feuille à n'importe quel stade de sa vie.

Nous utilisons de plus la distance par rapport aux nervures d'un pixel de la texture finale. En effet, la couleur de la feuille est directement liée à cette distance. La sève se retirant peu à peu de la feuille dans le réseau de nervures pour réintégrer l'arbre. Nous pouvons voir à la Figure 2.23 les différentes phases du vieillissement d'une feuille de tilleul. Avec une couleur jeune verte et une couleur vieille spécifiée comme étant rouge, nous avons généré les couleurs d'une feuille de tilleul tout au long de sa vie grâce au paramètre de mélange.

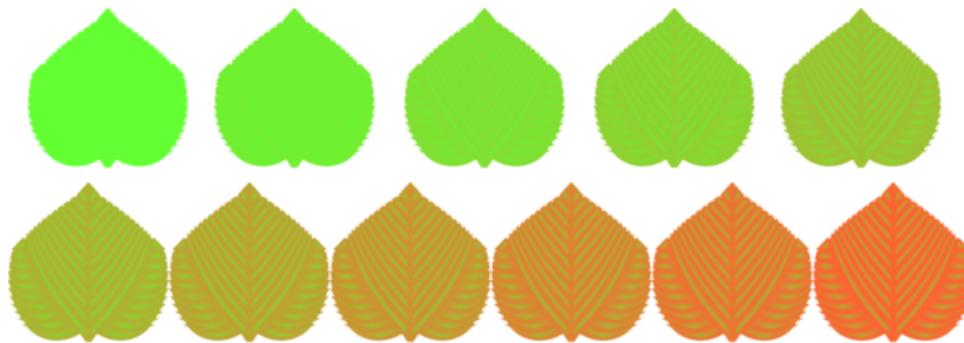


FIGURE 2.23 – Les couleurs d'une feuille de tilleul à différents âges. Nous avons utilisé le paramètre de mélange pour simuler le vieillissement de la feuille.

Pour donner un aspect plus réaliste au vieillissement des feuilles, nous générons un masque de mélange. Ces masques sont automatiquement générés pour simuler les zones de vieillissement sur une feuille. En effet, chaque zone va vieillir à un rythme différent de celui des zones voisines. De ce fait, ces masques nous permettent la génération de textures beaucoup plus complexes en autorisant le mélange de différents âges et ainsi d'avoir des âges locaux au sein d'une même feuille. Nous pouvons voir l'action d'un tel masque sur la Figure 2.24.

Deux couleurs sont choisies par l'utilisateur, une verte pour l'état jeune et une couleur jaune pour l'état vieux. Ces couleurs sont ensuite utilisées pour générer les textures de couleur en fonction de l'âge de la feuille, ce sont les deux textures à gauche de la Figure 2.24 en haut et en bas. Enfin, le masque de mélange (au centre à gauche) est appliqué pour donner deux âges sur deux zones différentes

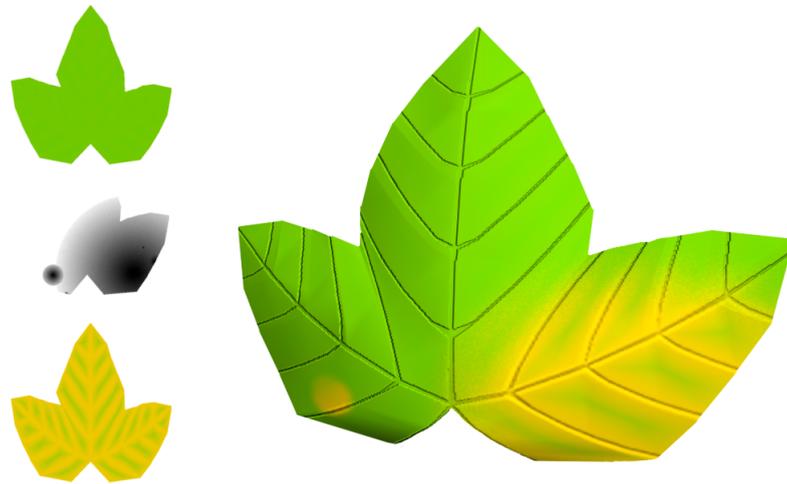


FIGURE 2.24 – L’action du masque de mélange sur la génération de la texture d’une feuille de lierre. Nous pouvons observer sur la gauche la couleur jeune, le masque de mélange puis la couleur vieille. Sur la droite la feuille texturée en utilisant le masque et les deux couleurs.

de la feuille. Ce masque peut par exemple simuler un ensoleillement hétérogène sur la feuille ou bien, à un niveau plus petit, l’action de gouttes d’eau concentrant la lumière du soleil sur certaines zones.

2.5 Résultats

Notre modèle permet de générer de grandes quantités de feuilles et de gérer leur vieillissement ainsi que les attaques d’insectes. Notre méthode, basée sur un 2-G-Carte L-Système paramétrique, offre la possibilité de définir facilement une grammaire puissante pour générer les nervures d’une feuille.

Le corps de cette feuille ainsi que son vieillissement sont ensuite générés automatiquement. Enfin, la possibilité est donnée à l’utilisateur d’effectuer le rendu des feuilles dans n’importe quel moteur de rendu externe. En effet, tous les éléments générés sont exportés sous forme d’un maillage et de textures.

La Figure 2.25 nous montre différents types de feuilles obtenues avec notre modèle de 2-G-Carte L-Système paramétrique. La grammaire est très efficace pour dessiner précisément la forme d’une feuille. Les blocs de scripts, les variables ainsi que les outils mathématiques mis à disposition permettent de représenter une grande variété de formes et de structures. Les grammaires utilisées pour la génération des feuilles présentées sur la Figure 2.25 sont ajoutées en annexes. Respectivement les grammaires Annexe .1, .2, .3, .4, .5, .6, .7, .8, .9, .10 et .11.

Nous pouvons voir sur la Figure 2.26 deux feuilles de liquidambar ayant des vieillissements différents. Ces deux feuilles montrent la gestion des masques de mélange avec ici trois couleurs utilisées. Les couleurs obtenues sont parfaitement en accord avec celles que l’on peut retrouver dans la nature. Les feuilles réelles à

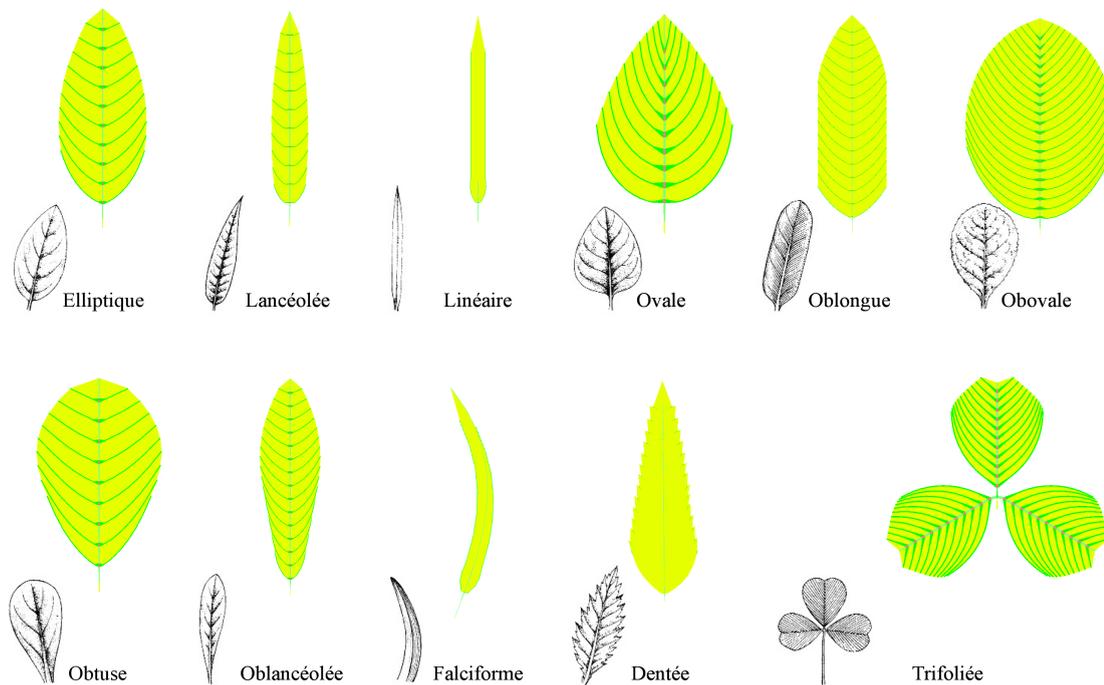


FIGURE 2.25 – Différentes formes de feuilles (schémas) ainsi que leurs représentations obtenues avec notre modèle.

droite sur la Figure 2.26 montrent un type de pigmentation que nous reproduisons de façon très réaliste avec nos masques de mélange.

La figure 2.27 illustre les différents niveaux de détail utilisés lors du peuplement de scènes naturelles. La scène de gauche est rendue sans utiliser de niveaux de détail, elle comporte 497610 polygones et a été calculée en 51 secondes sur un AMD Athlon 64 X2 5000+ avec 2Go de RAM. La scène de droite ne comporte que 338236 polygones et n'a pris que 28 secondes de calcul sur le même système avec l'utilisation de niveaux de détail. Même si la scène située à gauche possède moins de polygones que celle de droite, les deux résultats semblent pratiquement similaires.

Nous pouvons observer sur la Figure 2.28 les différentes étapes de notre méthode pour quatre espèces de feuilles. La colonne "Espèce" reprend une photo de la feuille que l'utilisateur souhaite modéliser. La colonne "Grammaire" montre le résultat des itérations de la grammaire. Nous pouvons remarquer que les formes de feuilles générées peuvent être très variées et posséder une ou plusieurs nervures principales.

La colonne "Génération du maillage et des textures" regroupe les différents outils générés pour effectuer le rendu des feuilles. Nous pouvons trouver en haut et de gauche à droite le maillage de la feuille, le masque comportant les attaques d'insectes puis les informations sur la position des nervures de la feuille. En dessous,



FIGURE 2.26 – A gauche, deux feuilles de liquidambar ayant subit des vieillissements différents. Pour cela nous avons utilisé des masques de mélange différents en gardant les couleurs choisies par l'utilisateur. A Droite des feuilles réelles ayant le même type de coloration

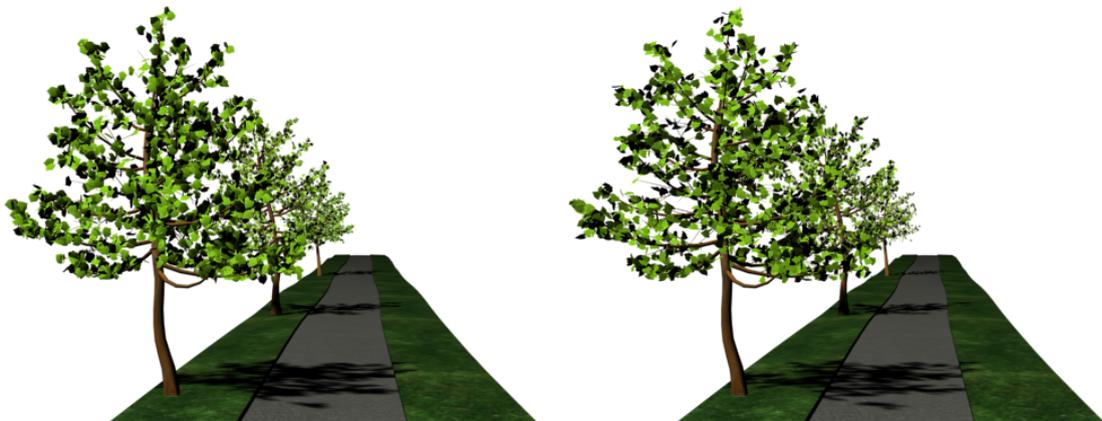


FIGURE 2.27 – Deux scènes similaires peuplées avec à gauche, uniquement des feuilles comportant beaucoup de détails et à droite des feuilles de moins en moins détaillées suivant l'éloignement sur la route. Les deux scènes semblent identiques.

nous avons les deux textures calculées en fonction des couleurs choisies par l'utilisateur et des informations sur les nervures puis le masque de mélange qui sera utilisé pour composer ces deux textures. Enfin dans la colonne " Résultats" nous pouvons observer le rendu final de la feuille en utilisant tous les éléments vus précédemment.

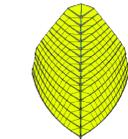
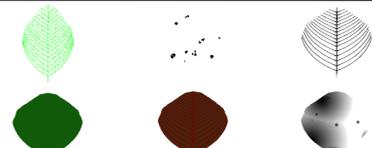
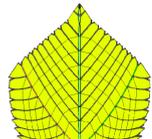
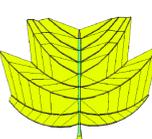
Espèce	Grammaire	Génération du maillage et des textures	Résultats
 Lierre			
 Rosier			
 Tilleul			
 Tulipier			

FIGURE 2.28 – Quatres espèces de feuilles modélisées avec notre méthode. Nous pouvons voir les photographies de feuilles servant de modèle, la génération de la grammaire et des textures puis enfin le rendu final de la feuille.

La Figure 2.29 nous montre l'importance d'inclure des différences dans la génération de feuilles de même espèce. Nous pouvons observer sur la gauche une tige de rosier possédant sept feuilles identiques alors que la tige de droite comporte sept feuilles différentes. La tige de droite nous parait beaucoup plus réaliste que celle de gauche du fait que ces feuilles sont différentes. La tige de gauche nous montre des feuilles complètement similaires avec une impression de copie trop présente. Celle de droite inclut quant à elle de petites différences presque imperceptibles consciemment qui ne donne pas cette impression.

Une tige de rosier comportant différents taux de vieillissement peut être vue sur la Figure 2.30. Nous pouvons y voir un vieillissement de plus en plus important dans le sens des aiguilles d'une montre. Les feuilles les plus jeunes ont une couleur vert tendre alors que plus leurs âges avancent plus leur couleur tend vers le rouge et se recourbent sur elles-mêmes. Les attaques d'insectes sont aussi présentes indépendamment du vieillissement des feuilles.



FIGURE 2.29 – A gauche une tige de rosier parfaite comportant sept feuilles identiques. A droite chaque feuille de cette branche est différente ce qui apporte beaucoup de réalisme à la plante.

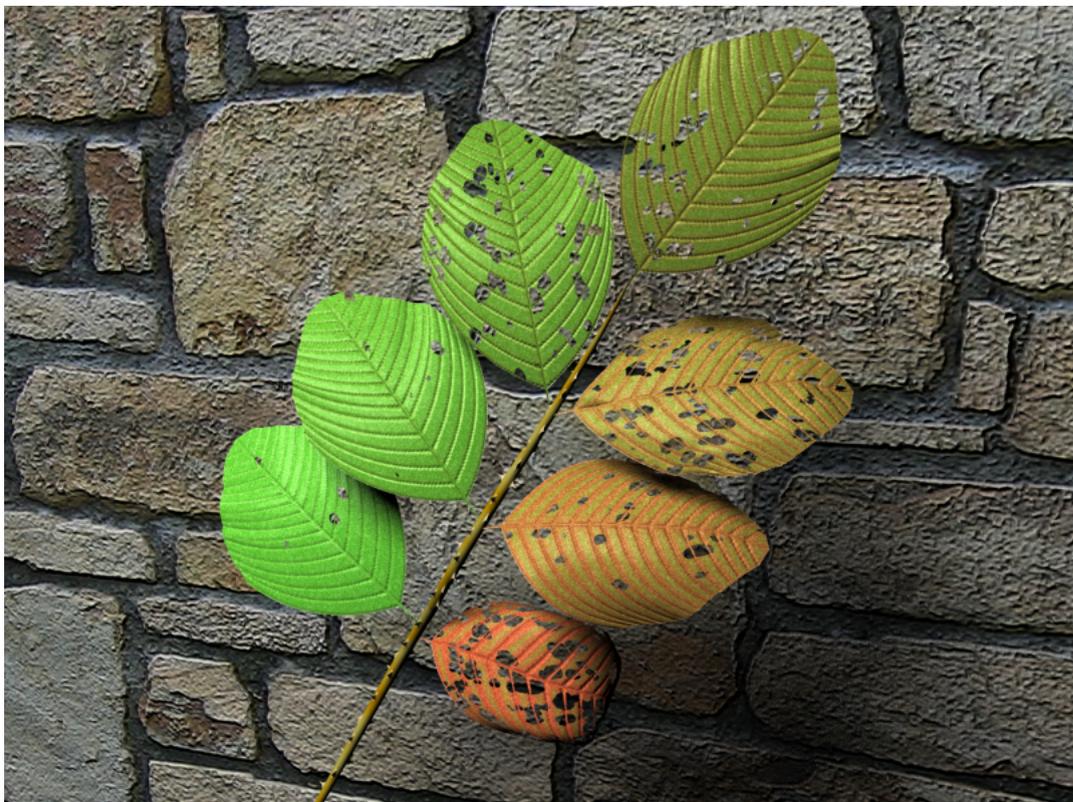


FIGURE 2.30 – L'illustration des différentes étapes du vieillissement d'une tige de rosier.

Les attaques d'insectes peuvent être plus ou moins violentes. La figure 2.31 nous montre sur la gauche des attaques d'insectes relativement importantes. Sur la droite, le limbe a été complètement détruit par des insectes, seules les nervures subsistent.

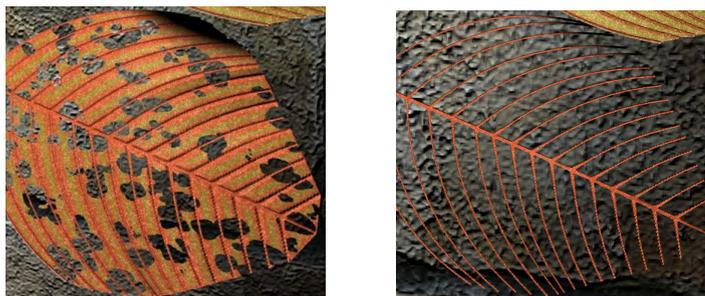


FIGURE 2.31 – Différents niveaux d'attaques d'insectes. A droite seules les nervures restent là où les insectes ont consommé le limbe de la feuille.



FIGURE 2.32 – Une branche de vieux lierre grimpant sur une poutre.

Nous pouvons observer sur la Figure 2.32 du lierre courant sur une poutre en bois. Ce lierre a subi un vieillissement de bas en haut. Nous pouvons voir au fur et à mesure le passage du vert au jaune, à l'orange puis au rouge. De plus, ses feuilles ont subi des attaques d'insectes sur toute la longueur de la clôture.

Le peuplement des arbres peut se faire pour différentes phases de vieillissement. Le résultat de ce peuplement est visible sur la figure 2.33 qui comporte quatre saisons différentes. La scène en haut à gauche nous montre le printemps avec des feuilles principalement jeunes et vertes. Sur la droite de cette saison, nous pouvons observer la fin de l'été avec des feuilles entre deux âges, entre vert et jaune. Sur la ligne inférieure, à gauche nous pouvons observer l'automne de cette scène. Les feuilles sont principalement rouges et jaunes et commencent à tomber sur le sol. Enfin, le début de l'hiver montre des feuilles très rouges qui ne tarderont pas à tomber de l'arbre pour se décomposer sur le sol.

Le nombre de feuilles différentes, tant au niveau de leur couleur que de leur forme, a une incidence directe sur le réalisme de la scène générée. Nous pouvons observer sur la Figure 2.34 trois images obtenues avec des variétés différentes. A gauche, 40 feuilles différentes sont générées pour peupler le tilleul. Au centre, 10 feuilles différentes sont utilisées. Enfin, sur la droite, seulement deux feuilles sont introduites dans la scène. Plus le nombre de feuilles différentes est élevé, plus le réalisme est important. Ces détails sont peu perceptibles consciemment, mais joue néanmoins un grand rôle pour le réalisme de la scène.

Chaque feuille possède une orientation particulière qui lui est

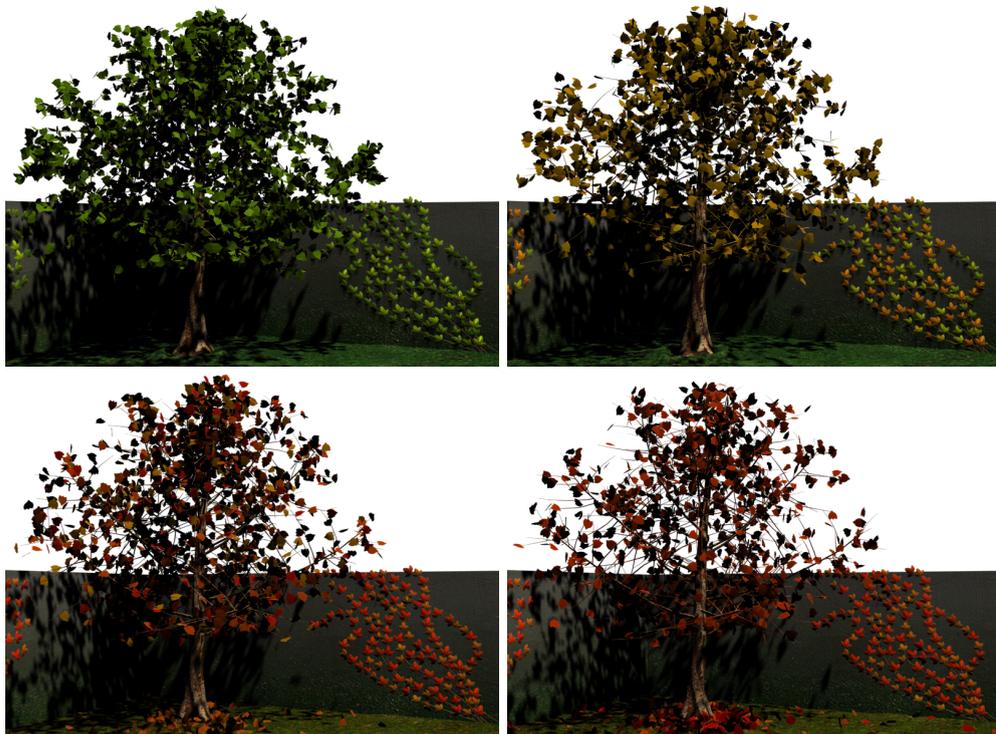


FIGURE 2.33 – Quatre saisons différentes pour un arbre et un plan de lierre. Le printemps, la fin de l'été, l'automne et l'hiver sont présent sur cette figure.

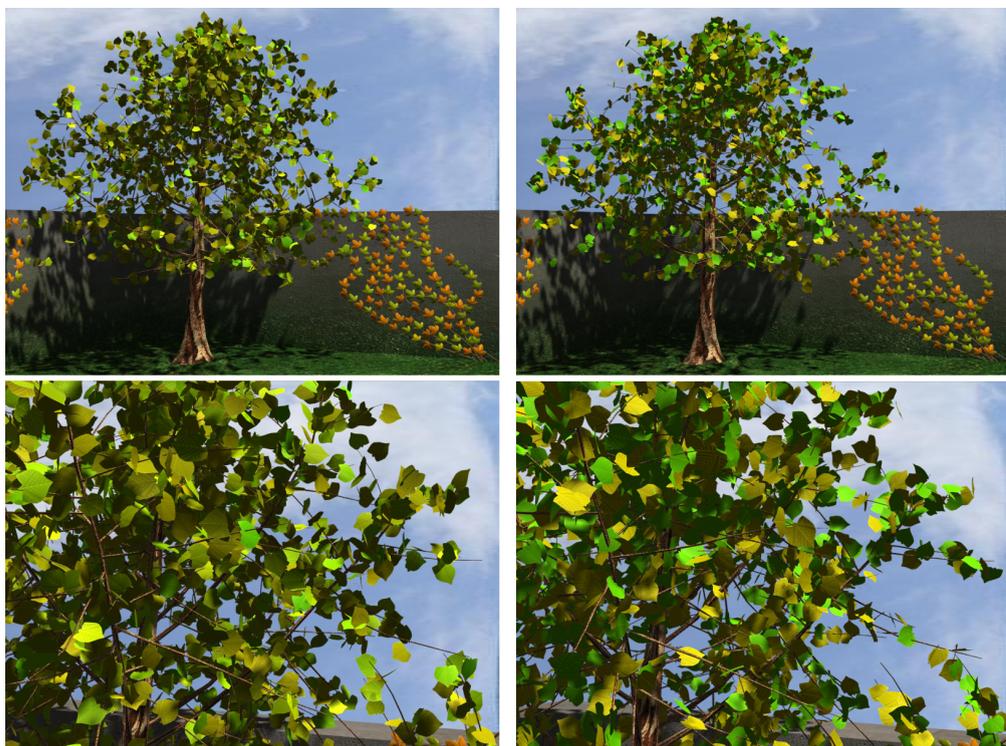


FIGURE 2.34 – L'utilisation de feuilles différentes impacte le réalisme de la scène. Ici 40 puis 2 feuilles différentes sont utilisées pour peupler la scène (de gauche à droite). Plus le nombre de feuilles différentes faible, plus le réalisme de la scène l'est aussi.

donnée à la création de l'axiome, c'est-à-dire la première face créée. Cette face déterminera le sens de la feuille, et donc comment il sera possible de l'inclure dans l'arbre. C'est cette orientation qui sera utilisée par la suite afin de positionner des feuilles sur un maillage représentant un arbre.

Pour tous ces arbres, les feuilles sont positionnées sur les branches suivant la même méthode. Tout d'abord, les facettes du maillage composant le tronc et les branches de l'arbre sont marquées. Une facette sera soit dans la catégorie tronc, soit dans la catégorie branche. Les facettes marquées comme faisant partie du tronc ne comporteront pas de feuille alors que les facettes catégorisées dans les branches pourront en porter. Pour chacune des faces pouvant porter des feuilles, des points aléatoires sont calculés sur ces faces puis des feuilles y sont positionnées suivant leur orientation. Cela nous assure que chaque feuille sera bien orientée par rapport à la facette porteuse.

2.6 Conclusion

Grâce à notre modèle, pratiquement tous les types de feuilles peuvent être modélisés. De plus, en s'aidant des blocs de scripts et des variables l'utilisateur peut facilement et rapidement créer une grammaire capable de générer une feuille complexe. La facilité de conception de la grammaire tient au fait que l'utilisateur ne doit que se concentrer sur les nervures principales de la feuille. Tous les éléments internes du limbe seront ajoutés automatiquement à chaque itération.

Nous avons de plus la possibilité de générer une grande quantité de feuilles toutes différentes, mais appartenant à la même espèce. Ces petites différences intra-espèce venant du fait que les feuilles ont toutes des conditions différentes de développement sont facilement intégrées dans la grammaire. De ce fait, toutes les feuilles générées avec une seule grammaire pour un arbre donné auront une forme différente ce qui participe grandement au réalisme de la scène.

Le vieillissement des feuilles se fait de façon naturelle grâce aux quelques paramètres fournis par l'utilisateur. Ce vieillissement, basé sur la botanique, permet de générer une feuille tout au long de sa vie. Cette génération gère les déformations géométriques telles que le séchage et la courbure des feuilles ainsi que les changements de couleur en tenant compte de phénomènes biologiques comme l'ensoleillement ou bien le flux de sève dans la feuille. De plus, notre modèle permet de générer un grand nombre de feuilles avec un âge différent pour une même espèce. Cela permet de peupler de façon plus réaliste un arbre au cours des saisons.

Notre méthode gère de plus les attaques d'insectes. Ces attaques peuvent survenir à différents endroits de la feuille, mais suivent quelques règles. Par exemple, le fait que les insectes ne s'attaquent que très rarement aux nervures des feuilles ou bien que les attaques suivent le déplacement des insectes. Notre modèle simule

des attaques pour différentes tailles d'insectes pouvant être en groupe ou bien solitaires. Finalement, les attaques peuvent avoir lieux à n'importe quelle phase du vieillissement de la feuille.

Grâce à tous ces traitements, une scène naturelle peut être facilement peuplée de feuilles appartenant à différentes familles, et ce, pour n'importe quelle saison. Ces feuilles seront toutes différentes suivant leurs formes et leurs taux de vieillissement ce qui apporte beaucoup de réalisme à la scène naturelle modélisée. Notre modèle permet de gérer un très grand nombre d'espèces de feuilles et, avec quelques paramètres demandés à l'utilisateur, de générer le feuillage complet de l'arbre désiré.

Chapitre 3

Génération d'empreintes et de traces sur des terrains

Sommaire

3.1	Introduction	97
3.2	problématique	98
3.3	Un modèle topologique	99
3.4	Génération de trajectoires	104
3.5	Enfoncement de la trajectoire et dépôt de matière	108
3.6	Résultats	117
	3.6.1 Extensions	122
3.7	Conclusion	124

Chapitre 3

Génération d'empreintes et de traces sur des terrains

3.1 Introduction

A une plus grande échelle de détails, les empreintes et les traces laissées sur le sol sont des éléments que l'on retrouve souvent dans des scènes naturelles. Ces détails sont présents sur une large gamme de tailles et de formes allant d'une empreinte laissée par un petit animal à de grandes traces imprimées par des roues d'engins. Que ce soit une activité humaine, animale ou bien géologique, il y a toujours des altérations laissées au niveau du sol. Ces traces peuvent être directement creusées dans un sol meuble ou bien imprimées sur un sol dur par dépôt et arrachement de matière. De plus, de nombreuses scènes naturelles sont concernées. Des traces et empreintes peuvent se retrouver dans la forêt, sur une plage ou bien sur un chantier comportant de la boue ou du sable. La principale caractéristique des déformations liées aux empreintes et traces sur un terrain est l'abaissement du niveau du sol. En effet, pour tout matériau meuble, la pression exercée par un objet sur celui-ci va le déformer et créer une empreinte. De plus, une certaine quantité de matière peut transiter par la surface de l'objet ayant réalisé cette empreinte. Sur tout le parcours de cet objet, nous pourrions donc observer de la matière déposée depuis, par exemple les pas précédents sur les pas suivants. De ce fait, la création de ces détails doit se faire en connaissant exactement la composition du sol en dessous de la déformation afin de pouvoir simuler l'enfoncement et le transport des matériaux composant le sol.

Les principaux objectifs des travaux présentés dans ce chapitre sont de modifier un sol en lui appliquant des empreintes ainsi que de gérer les dépôts de matières survenant à la suite de passages dans un sol meuble. La création de tels détails requiert une connaissance précise du sol à tout endroit de la trajectoire. Cette connaissance nous offrira la possibilité de gérer les matériaux sous-jacents afin de les

enfoncer et de les faire migrer le long de la trajectoire empruntée par l'objet en mouvement. De plus, le fait de connaître exactement les couches de matériaux composant le terrain à déformer permettra d'en modifier la structure de façon homogène. Dans cette optique nous avons développé un modèle de sol nous permettant de générer des pistes entières réalisées par des véhicules ou des marcheurs. Au moyen d'outils simples, l'utilisateur pourra définir des trajectoires d'empreintes respectant les formes qui seront fournies. Ces formes pourront par exemple être des empreintes de marcheurs ou bien des motifs laissés par la surface d'un pneumatique. L'utilisation de niveaux de détails permettra de générer un très grand nombre de détails sur n'importe quel terrain. Enfin, la matière qui sera arrachée lors des empreintes dans des sols meubles se verra déposée le long de la trajectoire, en aval du point de retrait.

Ces travaux ont été présentés dans un rapport de recherche [Pey11] décrivant les méthodes employées pour altérer et gérer des terrains soumis à de telles déformations.

3.2 problématique

Pour déformer un sol, il faut savoir quels sont les éléments qui le composent. Pour ce faire, nous utilisons un modèle de terrain donnant accès aux couches de matériaux sous-jacentes. Une représentation similaire du sol est utilisée dans [BF01] et [PGMG09]. Notre modèle n'a cependant pas besoin d'une résolution fixe. Il nous permet d'augmenter la définition du maillage localement, uniquement où nous avons besoin d'inclure des détails.

Une fois que les éléments composant le sol sont connus, l'utilisateur doit pouvoir définir où seront appliquées les traces et empreintes. Pour cela, nous lui proposons un outil permettant de définir précisément les trajectoires effectuées par les marcheurs ou véhicules désirés. Une trajectoire ne sera pas créée grâce à un objet animé comme dans [CLH96], [SBH99] ou bien [CLS07] qui utilise des modèles physiques. L'utilisateur fournira un chemin ainsi que la forme des empreintes qui seront utilisés le long de ce chemin.

Avec ces deux éléments, nous pouvons par la suite imprimer les empreintes en suivant le chemin fourni. Ces empreintes seront décalquées dans la structure du sol au moyen d'un raffinement puis enfoncées dans le sol en prenant le paramètre de hauteur des empreintes. La plupart des méthodes visant à effectuer des traces dans le sol le font au moyen d'un moteur physique. Notre but n'étant pas de procéder à l'animation de la formation de telles traces comme dans [ON03] par exemple, nous générons l'aspect final du sol uniquement.

A chaque empreinte imprimée dans le sol, nous connaissons la nature de matière sous-jacente. De ce fait, nous pouvons en déduire si de la matière est emportée

lors du contact avec le sol. Cette quantité de matière emportée est alors retirée des couches de matière du sol et stockée pour être déposée en aval de la trajectoire.

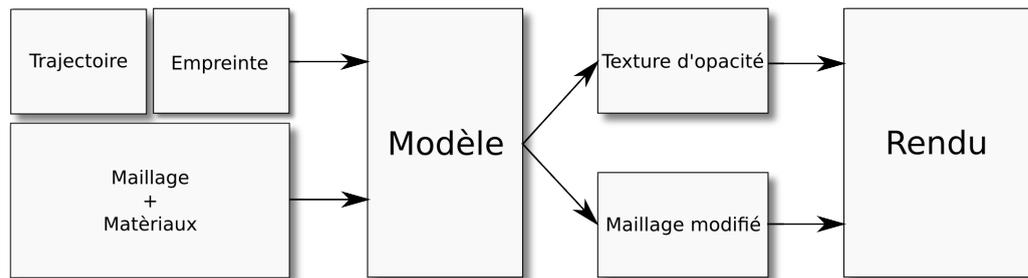


FIGURE 3.1 – Les données fournies à notre modèle sont utilisées puis des outils sont générés pour effectuer le rendu du terrain altéré.

La Figure 3.1 montre les différents éléments fournis par l'utilisateur dans le processus de modification du terrain. Une trajectoire, la forme de l'empreinte ainsi que le terrain à altérer sont utilisés par notre modèle. Une fois la trajectoire d'empreintes appliquée au terrain et les traces de matériaux déposées le long de la trajectoire, le modèle exporte le maillage modifié et une texture d'opacité détaillant où des dépôts de matière ont eu lieu.

Pour effectuer ces altérations, nous aurons besoin de plusieurs outils. Tout d'abord, notre modèle de terrains devra gérer la structure des matériaux sous la surface. Ce terrain devra pouvoir être construit facilement, à partir d'un maillage de base et des informations de matériaux. La résolution du terrain devra aussi pouvoir être localement augmentée pour pouvoir ajouter finement des détails. Ensuite, l'utilisateur devra pouvoir efficacement définir une trajectoire sur le terrain ainsi que la forme des empreintes. Ces formes vont effectivement modifier le sol en altérant la surface ainsi que la structure des matériaux en dessous. Enfin, les matériaux arrachés lors de l'estampage des empreintes le long de la trajectoire devront pouvoir être déposés au niveau des empreintes suivantes.

3.3 Un modèle topologique

Afin d'efficacement modifier un maillage, nous avons besoin de connaître rapidement les voisins des sommets et des arêtes. Cette fonctionnalité est offerte par les maillages traditionnels, mais très longue à obtenir (parcours exhaustif de tous les sommets). C'est pour cette raison que nous avons utilisé le modèle des 2-G-Cartes. Ce modèle topologique permet, grâce aux involutions utilisées, de connaître les voisins de n'importe quel élément, avec un coût minime.

En comparaison avec un maillage classique, notre modèle reposant sur une 2-G-Carte permet de gagner un temps d'exécution considérable comme le montre la Figure 3.2. Ici le temps de recherche exposé est en fonction du nombre de triangles présents dans la scène ainsi que du nombre de voisins à rechercher. La recherche avec notre modèle à base de 2-G-Carte voit son temps augmenter uniquement en fonction du nombre de voisins à rechercher alors que le maillage classique voit son temps de recherche de voisins augmenter de façon exponentielle.

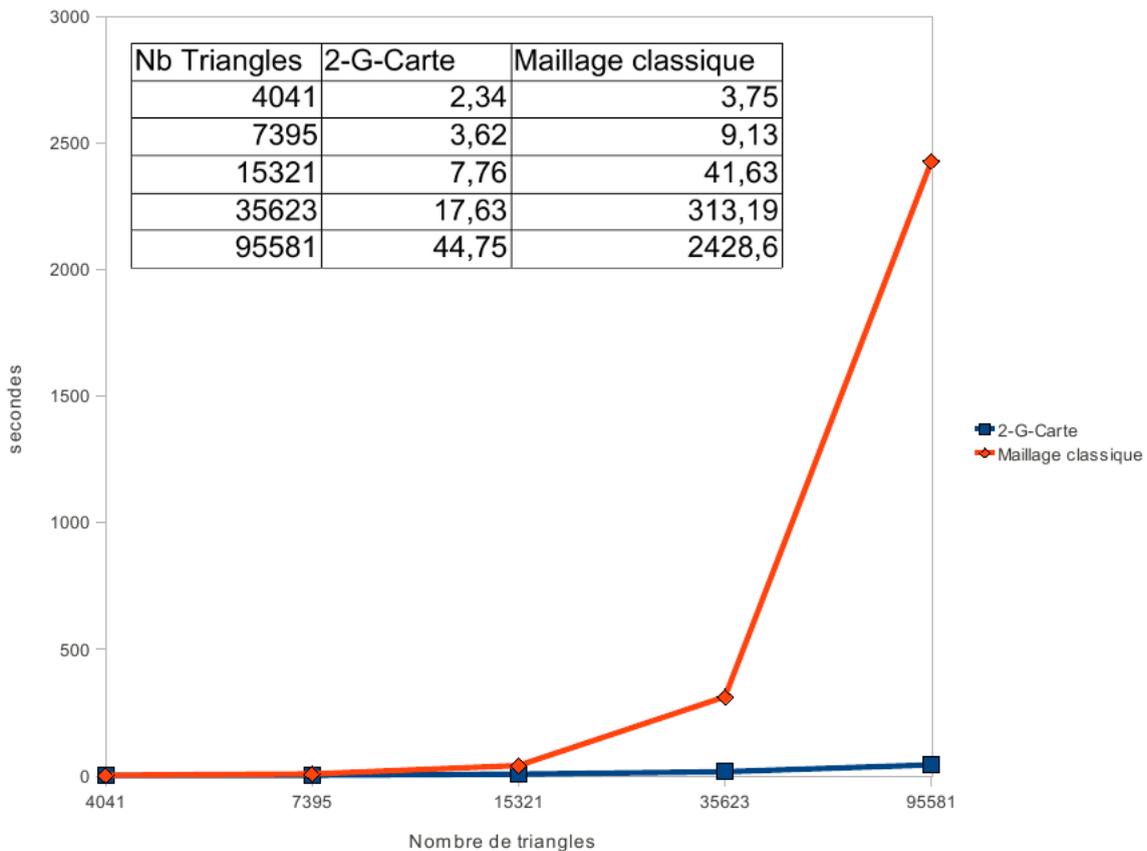


FIGURE 3.2 – Comparaison des temps de recherche de voisins. En bleu, notre modèle basé sur une 2-G-Carte. En orange, un maillage classique.

La description de ce modèle a été donnée dans la partie 1.1.1. A partir d'un brin modifié, nous obtenons facilement ses voisins qui pourront à leur tour être altérés. Nous pouvons observer sur la Figure 3.3 les involutions nous permettant d'obtenir la topologie.

Notre modèle inclut une nouvelle liaison β_1 qui permet de connaître les matériaux situés en dessous du terrain. Cette liaison donne accès à des couches de matériaux en rapport avec le brin. Nous pouvons voir sur la Figure 3.4 l'ajout de ces liaisons dans notre modèle.

Le terrain ainsi représenté nous offre la possibilité de connaître les voisins d'un élément particulier ainsi que la structure des couches de matériaux en dessous de chaque brin. Par la suite, lorsque nous ajouterons des détails dans le maillage,

nous aurons la possibilité d'interpoler les différentes couches de matériaux lors du raffinement pour obtenir une structure cohérente pour l'ensemble des couches du terrain.

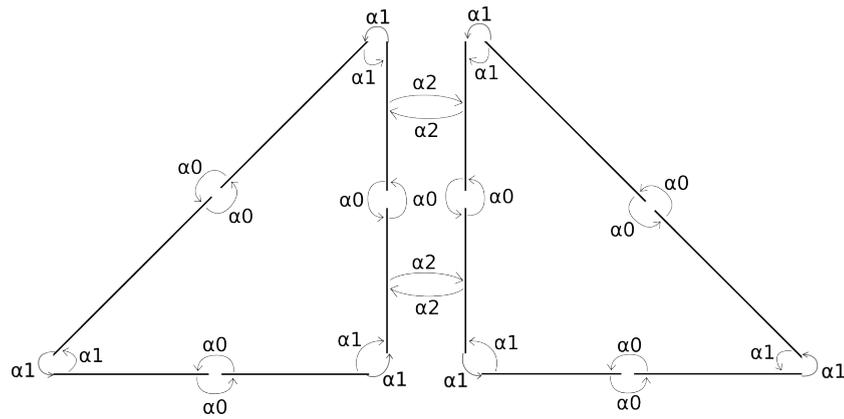


FIGURE 3.3 – Les différentes involutions mises en jeu dans le modèle des 2-G-Cartes. Ici nous obtenons deux triangles liés par une arête centrale.

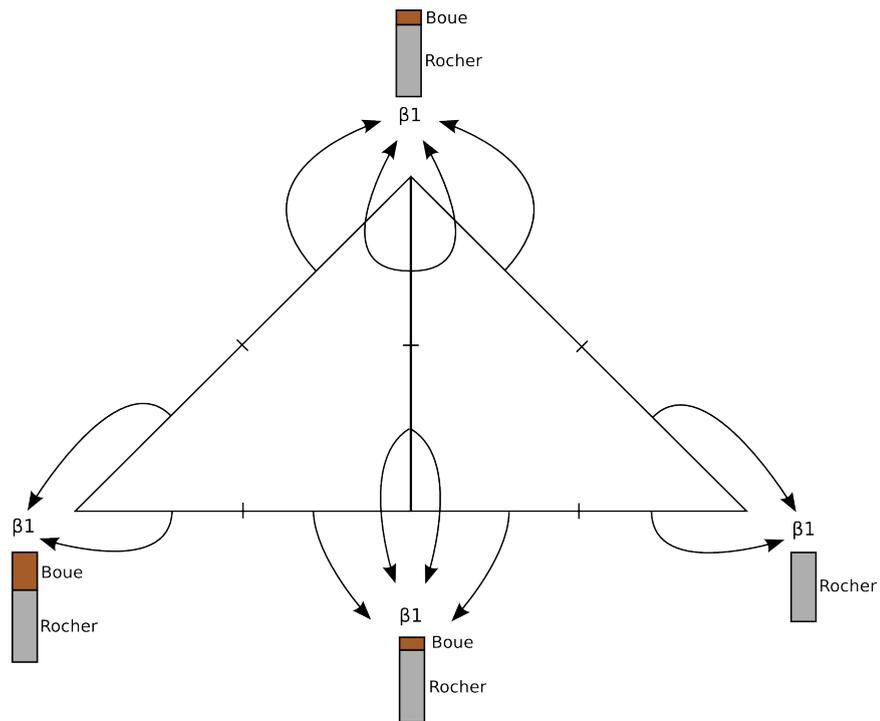


FIGURE 3.4 – Les liaisons β_1 nous permettent de lier la structure du sol sous forme de couches à chaque brin de la 2-G-Carte.

Outils topologiques du terrain :

Ce modèle de terrain est construit à partir d'un maillage de base auquel nous ajoutons des informations de matières pour chaque sommet. Nous avons la possibilité d'importer n'importe quel maillage triangulaire ainsi que les couches de matériaux qui lui sont associées. Pour effectuer cette opération, nous créons les coutures nécessaires à la 2-G-Carte ainsi que les liaisons avec les couches de matériaux. Les données géométriques concernant les sommets et les éléments géologiques de couches de matériaux sont exportées grâce à un script. Ce script s'occupe de récupérer les éléments modélisés et de les transformer pour pouvoir les interpréter dans notre modèle.

Une fois toutes les données converties, nous avons augmenté notre modèle avec la possibilité d'effectuer le raffinement des faces. En effet, nous voulons avoir la possibilité de modifier la résolution de la structure là où les empreintes seront appliquées au sol. Pour effectuer cette opération, nous nous sommes inspirés de la méthode classique PTQ (*Primal Triangle Quadrisection*). Nous devons cependant faire très attention aux données topologiques ainsi qu'aux couches de matériaux que nous avons ajoutées au maillage de base.

Nous pouvons voir sur la Figure 3.5 le résultat du raffinement topologique à partir d'un carré formé de deux triangles. Ces triangles sont raffinés afin d'obtenir pour chacun, quatre triangles de même taille. Les arêtes des triangles originaux sont divisées en deux pour obtenir les dimensions des nouveaux triangles. Pendant que ces divisions sont effectuées, les liens topologiques sont recréés pour garder la cohérence du maillage intacte. Ces liens topologiques induisent que les faces voisines d'une face subdivisée devront elles aussi être découpées en deux pour à la fois garder les liens, mais aussi pour restreindre la subdivision aux faces sélectionnées et leurs faces voisines. La figure 3.5 montre la cohérence des liens ainsi obtenus sur la droite.

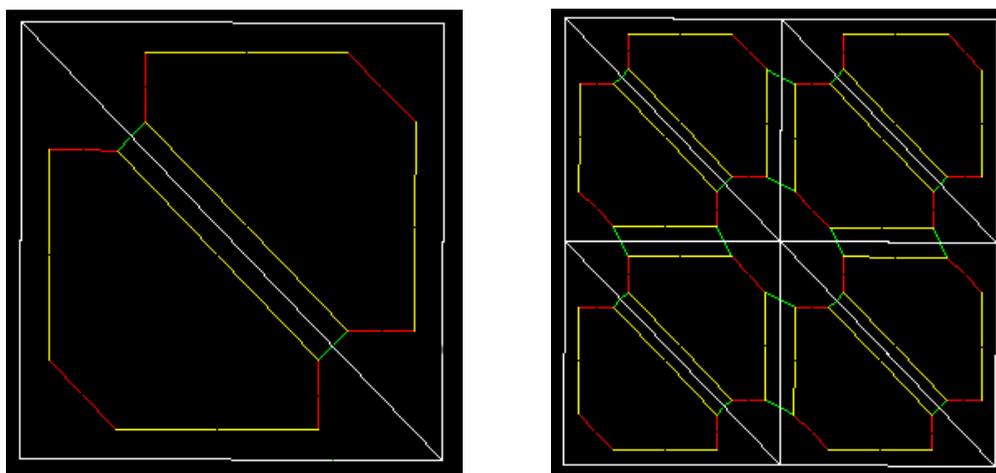


FIGURE 3.5 – Deux triangles sont subdivisés chacun en quatre triangles suivant le principe du PTQ. Nous pouvons observer sur cette figure en jaune les liaisons α_0 , en rouge les α_1 et en vert les α_2 .

Les couches de matériaux sont elles aussi sujettes au raffinement. En effet, lorsqu'un triangle est subdivisé, ses arêtes sont découpées en deux. Les brins ainsi créés doivent se voir affecter de nouvelles structures en couche définissant la structure des matériaux du sol. Pour procéder à la création de cette nouvelle structure, nous effectuons l'interpolation des couches présentes sous les brins voisins (Figure 3.6). Les nouvelles couches sont créées en fonction des couches déjà présentes et de leurs épaisseurs puis sont liées aux brins nouvellement créés.

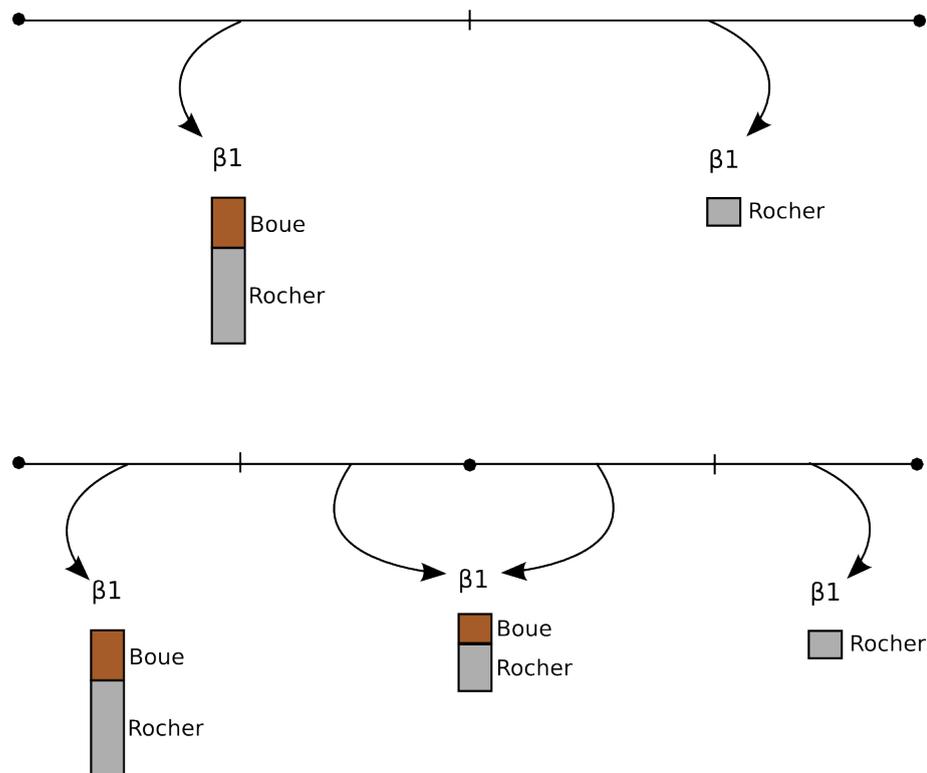


FIGURE 3.6 – Création et affectation de nouvelles couches pour les brins créés par raffinement d'un segment. En haut le segment initial avec à gauche une couche de boue surplombant une couche de roche et à droite une couche de roche à nue. En bas le segment après raffinement et création des nouvelles couches.

Avec cette représentation, la frontière, par exemple, d'une flaque de boue, sera présente sur le premier brin voisin ne comportant pas une couche de boue. Sur la Figure 3.6 la frontière de la boue est située sur le sommet de droite, là où la boue n'est plus présente. En effet, lors des raffinements successifs, des couches de boue de plus en plus minces seront créées jusqu'à arriver sur le sommet de droite ne comportant plus aucune trace de ce matériau.

Ce raffinement topologique sera ensuite utilisé pour ajouter des détails là où les empreintes agiront sur le sol. Le terrain pourra être plus ou moins raffiné selon les exigences de l'utilisateur et ainsi épouser la forme des empreintes tout en conservant la structure en couches du sol.

3.4 Génération de trajectoires

Une trajectoire est définie dans notre modèle sous la forme d'un chemin ayant de part et d'autre des empreintes. Ces empreintes peuvent être plus ou moins espacées et leur fréquence peut varier à cause d'une accélération ou bien une décélération de l'objet laissant des traces. Elles peuvent enfin varier en formes et en tailles en fonction du marcheur ou du véhicule qui les dépose. Pour réaliser de telles trajectoires, nous fournissons des outils interactifs afin que la génération soit la plus simple et la plus intuitive possible.

Définition du chemin :

Un chemin est défini par l'utilisateur en dessinant une ligne sur le maillage. Deux paramètres sont ensuite utilisés pour définir le nombre de pas et la largeur de la trajectoire. Ces deux paramètres couplés au tracé du chemin permettront de faire varier les trajectoires pour leur simuler une accélération ou une variation de vitesse. La possibilité est de plus laissée à l'utilisateur de faire commencer la trajectoire par un pas sur la gauche ou bien sur la droite. La Figure 3.7 nous montre trois trajectoires faisant varier ces paramètres. La trajectoire de gauche possède une forme basique, rectiligne, sans variations. Celle au centre montre la possibilité de dessiner des courbes complexes. Enfin, la trajectoire de droite représente une brusque accélération (de gauche à droite) avec un virage serré. Cette dernière peut représenter la trajectoire d'un athlète s'élançant pour une épreuve de saut en hauteur.

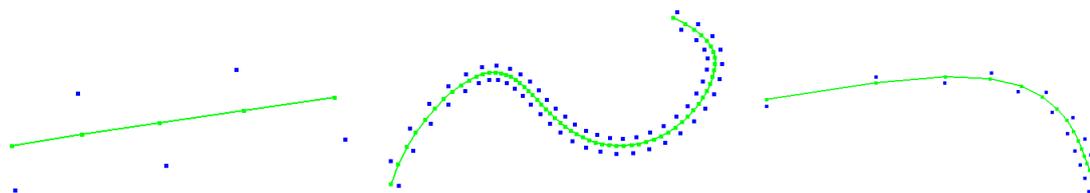


FIGURE 3.7 – Trois trajectoires dessinées par l'utilisateur. Nous pouvons observer les changements induits par la modification des paramètres.

Grâce à ces outils génériques, notre modèle offre la possibilité de définir des trajectoires complexes et pouvant comporter un grand nombre d'empreintes. Une fois le chemin tracé et les positions des empreintes calculées, il ne reste qu'à donner la forme des empreintes à appliquer à la trajectoire.

Définition des empreintes :

Les empreintes que nous utilisons pour altérer le sol comportent deux aspects prépondérants. Le premier est la forme générale de l'empreinte, cette information est fournie au modèle sous forme d'une texture représentant les parties de

l’empreinte qui entreront en contact avec le sol. Le second aspect réside dans la déformation du sol sous l’empreinte. Cette déformation sera donnée par une carte de hauteurs, définissant la manière dont l’empreinte s’enfoncera dans le sol.



FIGURE 3.8 – Différents types d’empreintes avec leurs cartes de hauteurs associées.

La Figure 3.8 montre différents types d’empreintes. Nous pouvons observer en haut à gauche une empreinte d’une patte de canidé ainsi que sa carte de hauteur correspondante. Cette carte de hauteur permet de faire s’enfoncer l’empreinte plus au centre des coussinets et ainsi rendre leur aspect bombé. De plus, les griffes ne sont pas adoucies pour rendre leur aspect dur marquant plus facilement le sol.

Sur la droite de cette empreinte nous présentons une empreinte de mouette qui à la particularité d’avoir des pattes palmées. Nous pouvons voir sur la carte de hauteur que la membrane constituant la palmure de la patte n’influe que très peu sur l’enfoncement. Seul le bord de cette palmure marquera plus profondément le sol.

La rangée du bas à gauche nous montre une empreinte de cheval ferré, nous pouvons voir que la carte de hauteur associée à cette empreinte est ne comporte pas de dégradés et est identique à l’empreinte. En effet, le fer du cheval étant constitué de matériaux durs, l’ensemble de l’empreinte sera déformé avec la même force.

Enfin, la dernière empreinte sur la droite nous montre une empreinte de pied, il est à noter que la forme de l’empreinte et sa carte de hauteurs ne possèdent pas le même contour. En effet, bien que de la matière puisse adhérer sous tout le pied, la partie interne ne marque que très faiblement le sol ici. Cette empreinte représente les traces laissées par un humain ayant des points d’appui normaux et marchant à une vitesse raisonnable.

Pour une trajectoire donnée, une seule empreinte est fournie. De ce fait, il faut appliquer alternativement cette empreinte et son symétrique de part et d’autre du

chemin. Cette symétrie se fait tout naturellement en sachant sur quelle position d'empreinte nous nous trouvons. Il suffit par la suite d'inverser les textures d'empreintes pour obtenir le bon résultat.

Pour créer des trajectoires d'empreintes plus complexes, il faut pouvoir fournir deux images représentant plus finement le type de trajectoire désirée. Par exemple, le fait de vouloir créer les empreintes laissées par un cheval au galop sera simulé en prenant directement une texture comportant les quatre pas d'un cheval parcourant une distance donnée au galop.



FIGURE 3.9 – Des textures d'empreintes nous permettant de simuler d'autres comportements de marche ou de course.

Nous pouvons voir sur la Figure 3.9 à gauche des traces de cheval n'ayant pas de fers. La carte de hauteur traduit cette absence de fers par le fait qu'elle laisse apparaître les différentes parties du sabot avec leurs différentes hauteurs. Ces empreintes visent à simuler le galop du cheval le long de la trajectoire. Les empreintes de droite représentent la course d'un lièvre bondissant avec ses pattes arrière pour retomber sur ses pattes avant.

Le dernier paramètre utilisé lors de la définition des empreintes est leur taille. En effet, aucun objet mobile ne laissera les mêmes traces qu'un autre sur le sol. Par exemple, des empreintes d'enfant et d'adulte, bien que correspondantes au niveau de la forme, seront de tailles différentes. De plus, cette taille dépend de l'échelle du terrain. L'utilisateur a le loisir de choisir la taille des empreintes qui correspondront le mieux à la trajectoire désirée. Ce paramètre permet d'accroître ou de diminuer interactivement la taille des empreintes sur la trajectoire.

Impression de la trajectoire :

Une fois toutes les informations de la trajectoire connues, les empreintes peuvent être appliquées sur le sol. Pour effectuer cette opération sur le sol, nous utilisons la texture donnant accès à la forme de l'empreinte. En suivant toute la trajectoire, nous sélectionnons les faces du modèle étant directement en dessous des empreintes pour ensuite être subdivisées. Toutes ces faces sont subdivisées en même temps pour éviter de découper les faces voisines en deux triangles lors de la subdivision topologique. Cette opération nous permet de produire une subdivision uniforme sur toute la trajectoire.

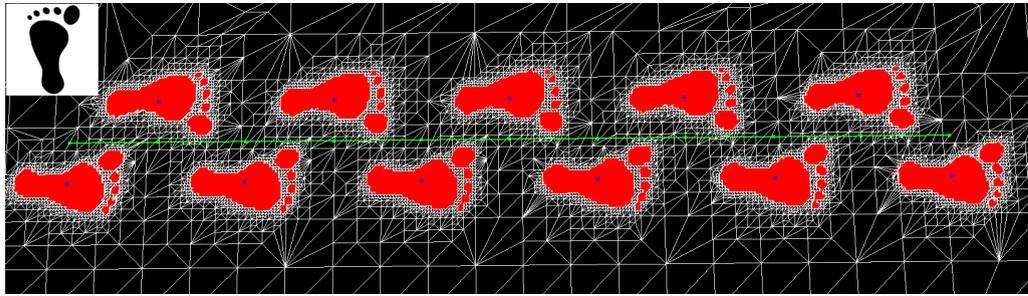


FIGURE 3.10 – L’impression d’une trajectoire sur le maillage. Nous pouvons voir en haut à gauche l’empreinte utilisée pour générer la trajectoire.

Pour cette étape nous utilisons la topologie de notre modèle, une fois toutes les faces sélectionnées, nous ajoutons aussi à la sélection les faces voisines pour obtenir un ensemble de faces englobant la trajectoire. Enfin, ces faces sont subdivisées et les phases de subdivisions se succèdent jusqu’à ce que le paramètre de raffinement maximum fixé par l’utilisateur soit atteint. La Figure 3.10 montre la génération d’une telle trajectoire. Elle comporte 11 empreintes imprimées sur le maillage. Nous pouvons voir que le maillage reste sensiblement uniforme.

Afin de réduire le nombre de triangles utilisés lors des subdivisions, le choix est laissé à l’utilisateur de ne raffiner que le bord des empreintes. Effectivement, pour une empreinte comme un pied, le niveau de détail au centre de l’empreinte n’est pas utile. Il est préférable de pouvoir définir plus finement le contour de celle-ci (Fig. 3.11).

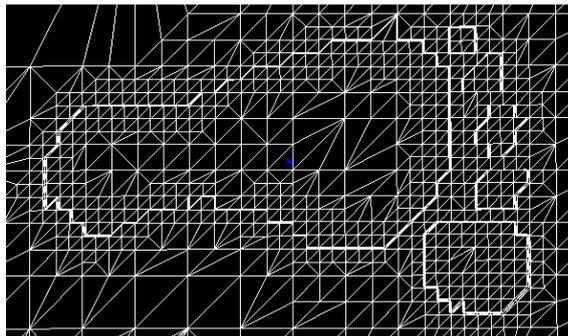


FIGURE 3.11 – Seul le bord de l’empreinte se voit ajouter des détails. Le centre change peu de résolution.

Notre modèle permet aussi la prise en compte des de niveaux de détails. Le long de la trajectoire, il est possible que des points assez éloignés de la caméra ne nécessitent que peu de subdivisions. En fonction de la position de la caméra, nous faisons varier automatiquement le paramètre de raffinement. Plus la caméra est éloignée d’une face à raffiner, moins la face sera subdivisée. Cela a pour effet d’ajouter des détails uniquement aux premiers plans de la scène. La figure 3.12 illustre cette amélioration en montrant deux points de vue différents de la même trajectoire.

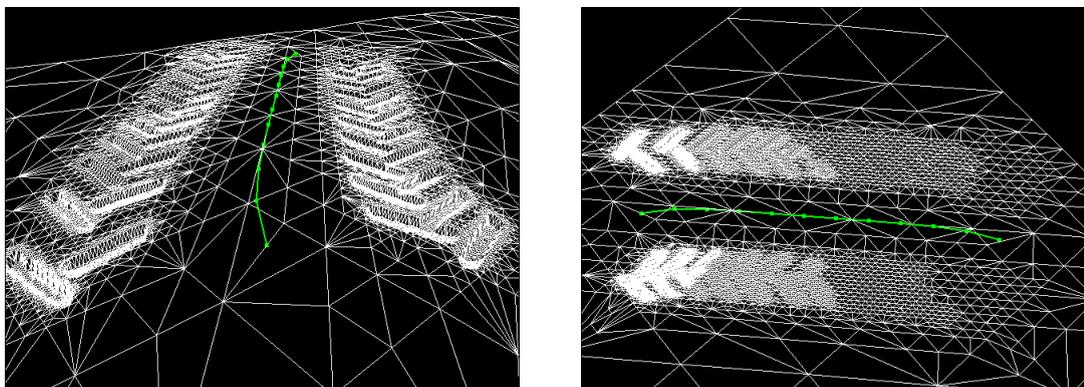


FIGURE 3.12 – A gauche, la vue de la caméra sur des traces de véhicules. A droite, les mêmes traces vues perpendiculairement. Nous voyons bien la gradation des détails en fonction de l'éloignement à la caméra.

Une fois les détails incrustés dans le maillage, il reste à enfoncer les empreintes suivant la structure du sol et les couches de matériaux présentes. Ce processus prendra en compte plusieurs paramètres tels que la vitesse ou le poids de l'objet laissant des traces sur le sol. Enfin, les dépôts de matière seront gérés lors de chaque interaction entre le sol et les empreintes.

3.5 Enfouissement de la trajectoire et dépôt de matière

Pour modifier la structure du sol et les couches sous-jacentes en fonction des empreintes fournies par l'utilisateur, nous avons besoin de paramètres génériques qui seront appliqués lors de l'enfoncement des traces dans le sol. La structure du sol sera abaissée en prenant en compte la carte de hauteurs fournie ainsi que les paramètres d'enfoncement. Cet enfoncement sera effectué en fonction des couches de matériaux présentes dans notre modèle pour décrire la structure interne du sol. Ces couches verront leurs hauteurs varier en fonction de la déformation appliquée au sol.

Une fois la déformation d'enfoncement effectuée sur l'empreinte, notre modèle permet de gérer des dépôts et retraits de matière. En effet, les matériaux composant le sol peuvent être assez adhérents aux objets entrant en contact avec eux. La boue est un bon exemple de ces types de matériaux, car elle possède une adhérence assez importante et transite sur de grandes distances lorsqu'un véhicule ou un marcheur y passe. Les retraits de matière sont alternés avec des dépôts qui nous permettent de simuler l'adhérence du matériau emporté le long de la trajectoire. Ces dépôts sont gérés grâce à des tampons représentant une partie de l'empreinte qui permettent de disperser la matière le long de la trajectoire et d'obtenir un effet réaliste de dépôts.

Enfoncement des empreintes :

Nous partons du point où, dans notre modèle, les faces correspondantes aux empreintes ont subi un raffinement topologique. Nous allons utiliser plusieurs paramètres afin de simuler l'enfoncement des empreintes dans un sol meuble.

Pour effectuer l'enfoncement dans le sol, notre modèle possède deux méthodes. Soit le bord des empreintes sera enfoncé de façon franche avec la création d'un rebord, soit les bords de l'empreinte seront doux et sans rebord. La première méthode repose sur une extrusion topologique qui permet de reconstituer les bords des empreintes. La seconde méthode, quant à elle, permet de simuler des rebords doux comme ceux présents sur les empreintes effectuées sur un sable sec ou bien sur un sol relativement souple.

Le poids de l'objet en mouvement sur le sol est le premier paramètre pris en compte. En effet, C'est la principale grandeur qui va nous permettre d'altérer le sol sur la hauteur. Plus l'objet est lourd, plus les traces qu'il engendre seront profondes dans le sol.

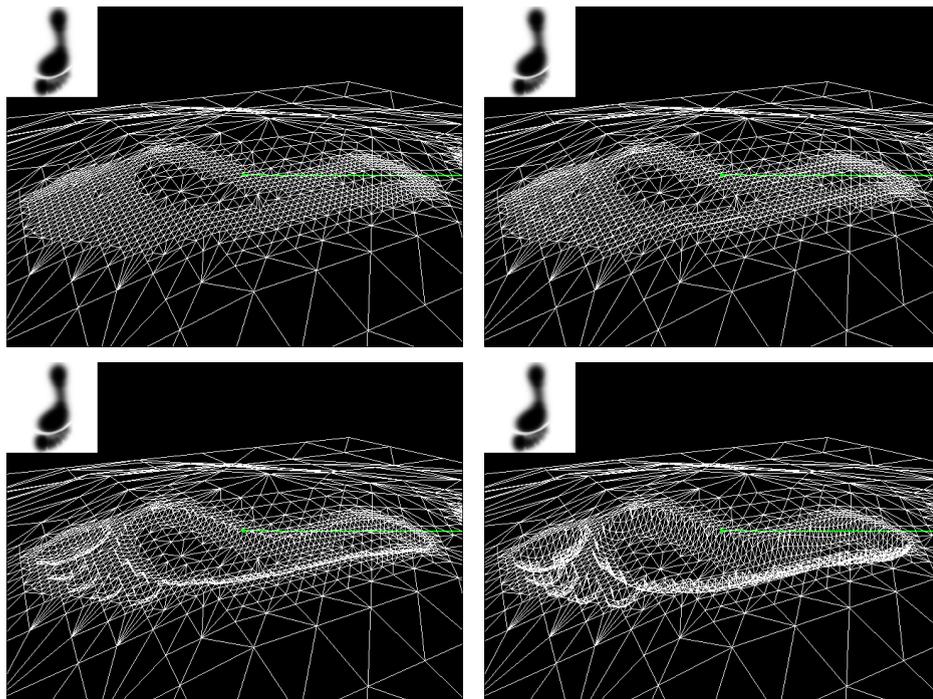


FIGURE 3.13 – Enfoncement d'une empreinte de pieds en fonction du poids. Plus le poids augmente plus l'enfoncement est conséquent. Nous pouvons voir une vignette en haut à gauche de chaque image représentant la carte de hauteur utilisée pour l'empreinte de pied.

Nous pouvons observer sur la Figure 3.13 le phénomène d'enfoncement plus ou moins prononcé suivant le poids du sujet. Nous pouvons de plus observer dans le coin supérieur gauche de chaque image une vignette montrant la carte de hauteur utilisée. Ici une empreinte de pied est mise en rapport avec la forme du pied d'un marcheur moyen. Cette carte de hauteur prend en compte les points d'appui d'un

marcheur ainsi que la forme de son pied.

La première image en haut à droite nous montre une simulation effectuée avec un poids nul. Bien que le raffinement de l'empreinte soit présent sur le maillage, aucune déformation en hauteur n'est appliquée. Les images suivantes font apparaître des empreintes de plus en plus profondes en fonction de la valeur choisie pour le poids du sujet. Ce poids permet de marquer plus ou moins intensément le sol grâce à la carte de hauteur fournie. Enfin, la Figure 3.13 nous permet de voir que seule l'empreinte est affectée par l'enfoncement.

Afin de gérer les rebords de l'empreinte, nous laissons à l'utilisateur le choix de la méthode utilisée. Soit le choix sera fait d'utiliser des bords doux pour ne pas faire apparaître de coupure franche entre l'empreinte et le reste du sol, soit les bords de l'empreinte seront reconstruits grâce aux informations topologiques de notre modèle.

La première méthode (sans bords francs) va abaisser les sommets du maillage en fonction de la carte de hauteur créée à partir du tampon d'empreinte considéré. Cette carte de hauteur possède des valeurs moins importantes à mesure que l'on s'approche des bords de l'empreinte. Ces valeurs deviennent nulles sur les bords proprement dits de l'empreinte pour obtenir une marque douce dans le sol altéré. De ce fait, l'enfoncement de l'empreinte dans le sol se fera de manière douce avec un bord qui correspondra au niveau du sol autour de l'empreinte.

Pour la méthode utilisant des bords tranchés n'ayant pas la même hauteur que le reste du terrain nous avons utilisé une reconstruction topologique des bords de l'empreinte. Notre méthode consiste en une extrusion des arêtes disposées sur les bords de l'empreinte puis en une couture des faces ainsi créées avec les faces existantes. Pour ce faire, nous utilisons les données topologiques de notre modèle afin de n'avoir aucun temps perdu en recherche de faces.

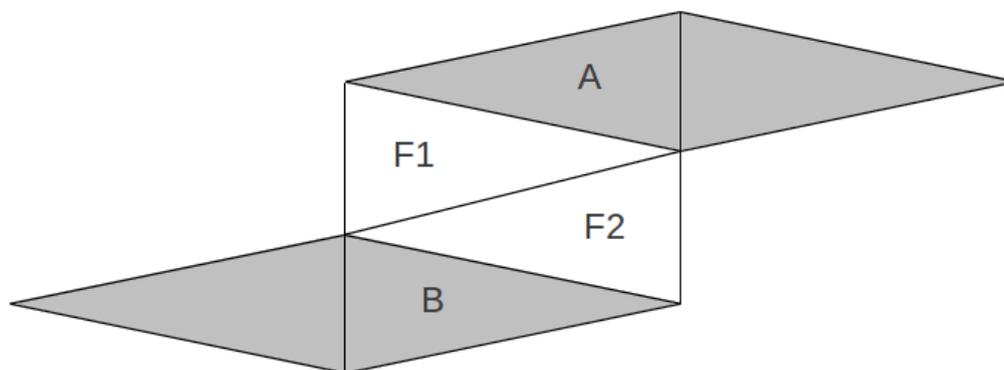


FIGURE 3.14 – Extrusion topologique des bords d'une empreinte dans notre modèle. Ici la face A est B sont reliées par un rebord constitué des faces F1 et F2.

Par exemple, sur la Figure 3.14, nous pouvons observer deux faces *A* et *B* représentant un morceau du bord d'une empreinte. La face *A* se situe du côté extérieur de l'empreinte alors que la face *B* quant à elle se trouve sur le côté intérieur

de l'empreinte. Lorsque notre procédure d'enfoncement arrive sur la face B , nous détectons facilement qu'il s'agit d'une face appartenant au bord de l'empreinte. En effet, avec la topologie, il suffit de regarder à quel groupe les faces voisines appartiennent. Si toutes les faces voisines sont dans le groupe de l'empreinte, alors il n'y aura pas de traitement particulier. Cependant, si une ou plusieurs faces n'appartiennent pas à l'empreinte, il nous faut effectuer une extrusion afin de recréer le bord intérieur de l'empreinte.

Cette extrusion se fait en deux temps, premièrement les sommets qui vont prendre part à ce processus sont enregistrés et la face appartenant à l'empreinte est enfoncée comme s'il s'agissait d'un enfoncement classique. Ensuite, les faces support du bord de l'empreinte sont créées en fonction des faces existantes. Afin de créer ces faces, les faces existantes (A et B sur notre exemple de la Figure 3.14) sont utilisées comme support. Des faces sont alors créées pour combler la faille qui a été formée lors de l'enfoncement de la face appartenant à l'empreinte. Ces faces sont ensuite cousues entre elles puis aux faces existantes pour obtenir un maillage cohérent. Qu'il s'agisse d'un cas comportant une face voisine en dehors de l'empreinte ou bien plusieurs, la seule différence sera une couture de plus entre les faces nouvellement créées sur le rebord intérieur de l'empreinte.

Les traces sont aussi très dépendantes des zones d'appui qu'un marcheur met en évidence lors de sa marche. Ces zones sont plus ou moins similaires suivant les individus et suivant les chaussures qu'ils portent. Ces deux derniers aspects sont facilement simulable en utilisant la carte de hauteur et en y injectant de la hauteur aux endroits où l'appui sera plus important. Cependant, plus le marcheur se déplacera vite, plus ces points d'appui seront déportés vers l'avant de l'empreinte. A l'inverse, un coureur freinant verra ses empreintes plus marquées sur l'arrière, car il utilisera fortement ses talons pour ralentir sa course. Nous avons donc inclus dans notre modèle un paramètre permettant de prendre en compte la vitesse de déplacement d'un marcheur.



FIGURE 3.15 – La carte de hauteur de l'empreinte est plus ou moins altérée en fonction de la vitesse. Ici la vitesse du sujet augmente de gauche à droite.

Pour effectuer cette modification des points d'appui vers l'avant ou l'arrière de l'empreinte, nous gérons un repère local pour chaque empreinte. Ce repère est calculé en fonction de la trajectoire en prenant le vecteur entre deux points adjacents de la trajectoire comme axe X , l'axe Z quant à lui ne diffère pas de l'axe Z globale et l'axe Y est déduit des deux autres axes. Ce repère nous permet donc de con-

naitre la place qu'occupe un sommet au sein de l'empreinte. Plus son abscisse sera importante et plus le sommet se trouvera vers l'avant de l'empreinte. De ce fait, notre modèle pourra appliquer un coefficient régulateur en fonction de la place du sommet dans l'empreinte. Ce coefficient permettra d'accentuer l'enfoncement soit vers la pointe de l'empreinte si le marcheur est phase d'accélération soit vers l'arrière de l'empreinte si le marcheur ralentit sa course. Nous pouvons voir sur la figure 3.15 l'altération de la carte de hauteur d'un marcheur pieds nus, cette carte de hauteur est altérée par la vitesse qui est croissante de gauche à droite. Plus la vitesse est élevée, plus les niveaux de la carte de hauteur sont estompés vers l'arrière de l'empreinte.

Nous nous sommes dans un premier temps focalisés sur des empreintes représentant un cas complexe, les empreintes de pas. Cependant, nous avons maintenant la possibilité d'aborder d'autres types d'empreintes telles que les traces laissées par un pneu de véhicule. Pour cela un autre paramètre est ajouté à notre modèle pour nous permettre de gérer efficacement la simulation d'une trajectoire faite par un véhicule sur le sol est pris en compte. Ce paramètre nous permet de décrire la taille des roues. Cette taille n'est pas le diamètre de la roue à proprement parler, mais le nombre de motifs présents à la surface de la roue. Ce nombre de motifs permet de connaître exactement la capacité de la roue à transporter la matière le long de la trajectoire. Nous pourrions ainsi savoir précisément quelle partie de la roue retiendra ou non de la matière. De plus, la matière sera déposée au sol en fonction du fait qu'il y ait de la matière ou pas sur le motif courant.

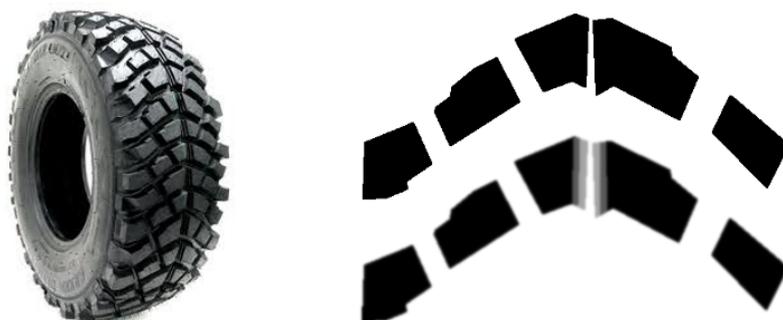


FIGURE 3.16 – A gauche, un pneu comportant plusieurs motifs se répétant sur sa surface. Nous pouvons voir sur la droite l'empreinte déduite ainsi que sa carte de hauteur.

Nous pouvons voir sur la Figure 3.16 un pneu simulé par notre modèle. Sur la gauche nous pouvons observer le pneu avec ses motifs se répétant sur la surface entrant en contact avec le sol. Sur la droite nous voyons l'empreinte déduite des motifs de ce pneu (en haut) et la carte de hauteur lui correspondant. Cette carte de hauteur permet de marquer les deux crans présents au centre du pneu.

Le dernier aspect de l'enfoncement des empreintes dans le sol se situe au niveau de la modification des couches de matériaux. Comme nous l'avons vu précédemment (Figure 3.4), notre modèle possède pour chaque brin une pile de couches

de matériaux lui étant liée. A chaque fois que le sol est enfoncé pour y laisser une empreinte, cette structure de couche est modifiée afin de savoir s'il y a la possibilité de s'enfoncer plus avant dans le sol ou non. Si la profondeur restante de matériaux meuble est insuffisante, la valeur de l'enfoncement sera diminuée pour correspondre à la valeur maximale de matière restante en profondeur sur la pile de matériaux. Ainsi, lorsqu'il n'y a plus de matière dans la structure en couche de matériaux, nous procédons à l'arrêt de l'enfoncement comme pour l'arrivée sur une couche de matière résistante et indéformable.

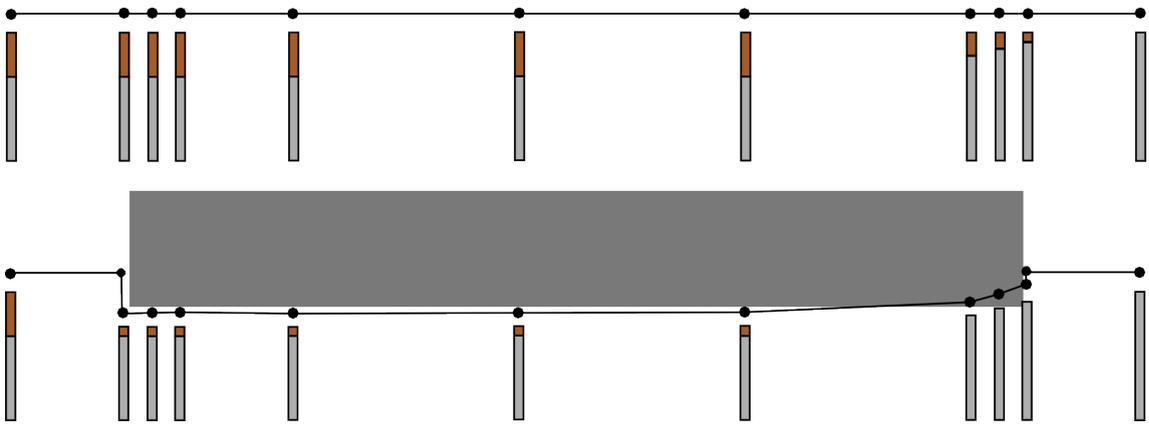


FIGURE 3.17 – Enfoncement d'une empreinte en tenant compte des couches de matériaux. En haut la structure du sol intacte, en bas la même structure altérée par une empreinte plate.

Nous pouvons observer sur la Figure 3.17 un profil d'enfoncement dans un sol. Notre modèle gère les couches de matériaux situées en profondeur dans le sol, ici nous sommes en présence du contour d'une flaque de boue. Nous voyons la hauteur des couches de boue (en marron) diminuer à mesure que l'on s'approche du bord (à droite). Sur la figure du bas, un enfoncement simple est effectué à l'aide d'une forme simple. Nous pouvons voir que sur la gauche de l'enfoncement, la matière nécessaire à l'enfoncement est présente. Cependant, sur la droite de l'enfoncement, il n'y a pas assez de matière. A cet endroit, le sol ne sera pas plus enfoncé que la hauteur des couches grises représentant une matière indéformable. Ce traitement simule que l'enfoncement est stoppé lorsque l'objet effectuant l'empreinte entre en contact avec de la roche (en gris). Nous pouvons de plus observer sur cette figure la création des bords de l'empreinte de part et d'autre du motif enfoncé dans le sol avec l'ajout de sommets supplémentaires.

La figure 3.18 montre le cas où l'empreinte est à cheval sur une surface indéformable et un sol meuble. Nous pouvons voir que le béton indéformable ne subit aucun enfoncement alors que la terre du monticule est enfoncée. Ce processus est automatique en fonction des couches de matériaux présentes dans le modèle.

Afin de conférer un aspect plus réaliste aux enfoncements des empreintes dans le sol, nous offrons la possibilité à l'utilisateur d'altérer les bords de l'empreinte de façon à les élever. Ce traitement permet de simuler le déplacement de matière



FIGURE 3.18 – Enfoncement d'une empreinte sur le bord d'un monticule de terre. Nous pouvons observer l'enfoncement sélectif qui s'opère sur la terre et non sur le béton soutenant le monticule.

migrant de l'empreinte à ses bords sous l'effet de la pression du sujet altérant le sol. Cette élévation des bords de l'empreinte se fait relativement simplement et sans coup de calcul particulier grâce à la topologie de notre modèle. En effet, les sommets considérés sont facilement atteignables en suivant les bords de l'empreinte. Nous pouvons observer ces résultats sur la Figure 3.24 où les bords des empreintes sont élevés. Nous pouvons voir deux résultats différents. Le premier (en haut à gauche) montre le résultat de l'élévation des bords avec une empreinte douce, cette trajectoire simule le passage de la mouette sur un sable relativement sec. En bas à droite la même élévation des bords est effectuée, mais avec des empreintes possédant des bords francs. Ce dernier résultat simule un sol plus dur correspondant à du sable humide.

Transfert de matière sur la trajectoire :

Notre modèle propose de simuler le transport de la matière qui pourrait rester collée à l'objet déformant le sol. En effet, de la matière peut être présente sous un pied ou une roue et de ce fait être transférée sur l'ensemble du chemin effectué par le mobile altérant le sol. Ce transport de matière se fait tout au long de la trajectoire en retirant et déposant de la matière sur les zones de contact entre l'objet et le terrain considéré. Pour définir les zones où la matière sera enlevée et déposée, notre modèle va faire appel à une collection de tampons générés à partir de l'empreinte source pour effectuer ces retraits et dépôts de matériaux sur le sol. Ces tampons seront ensuite utilisés avec une texture permettant de stocker les échanges de matière le long de la trajectoire. Une fois ces sommets connus, il suffit de leur appliquer la déformation voulue.

Chaque tampon sera utilisé pour déposer ou bien retiré de la matière sur

chaque empreinte. En effet, les tampons sont définis en fonction de l’empreinte. Leurs formes resteront donc contraintes à l’intérieur de la forme originale de l’empreinte considérée. Pour ce faire, l’empreinte est découpée aléatoirement pour donner autant de tampons que nécessaire. Un grand nombre de tampons générés assurera une simulation plus hétérogène au niveau de la forme des retraits et des dépôts. Ces découpages peuvent être effectués suivant les empreintes fournies, il suffit d’adapter le processus de découpe aux différentes formes utilisées.

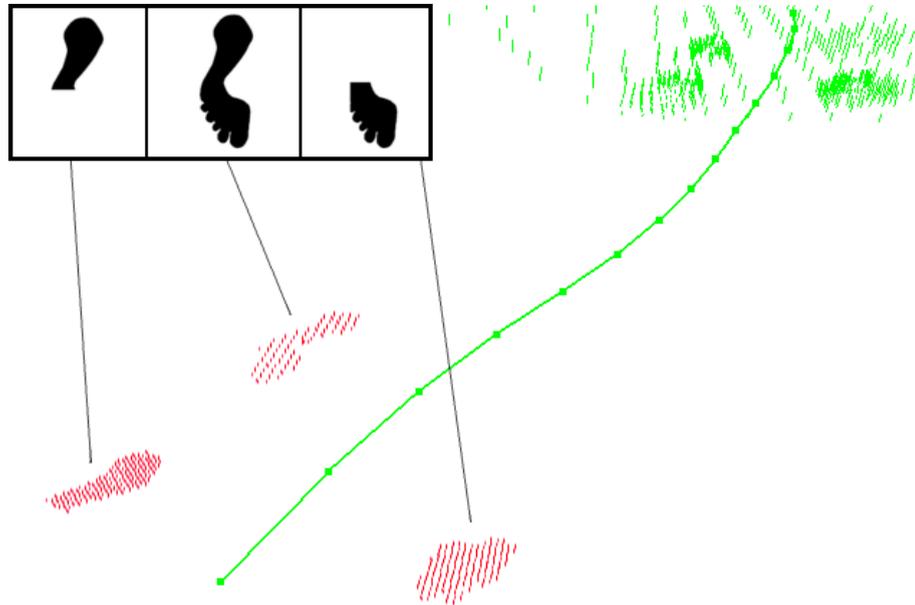


FIGURE 3.19 – Ici, trois tampons simples sont utilisés pour gérer les dépôts de matière. Nous pouvons voir en haut à gauche les trois tampons générés et en rouge les endroits où la matière est déposée ainsi qu’un lien vers le tampon responsable du dépôt.

Nous pouvons observer sur la Figure 3.19 en haut à gauche trois tampons générés de façon aléatoire. Ces tampons sont ici générés simplement. En effet, une découpe horizontale est effectuée sur l’empreinte pour donner une partie qui sera utilisée comme tampon. Nous pouvons voir de plus la trajectoire utilisée pour notre simulation. Les traits verts représentent les piles de matière présentes naturellement sur le terrain et les traits rouges représentent de la matière déposée lors de la simulation de dépôts le long de la trajectoire. Nous pouvons voir que la matière est déposée sur un sol dur (pas de présence de traits verts donc pas de matière déformable et transportable) et que le dépôt correspond aux tampons que nous montrons en haut à gauche de la figure. La matière est donc bien déposée suivant les tampons définis à partir de l’empreinte de base (ici un pied similaire au premier tampon présent sur la figure 3.19).

Pour gérer le transport de matière tout au long de la trajectoire, nous utilisons des textures dans lesquelles nous allons stocker la matière en transit sur le terrain. Ces textures vont représenter les surfaces des objets mobiles altérants de sol se

trouvant en contact avec le terrain. Le paramètre de la taille des roues nous est utile ici pour savoir combien de textures vont être utilisées pour simuler le passage d'une roue sur le terrain. Nous considérons ici que chacune des nouvelles empreintes pourra enlever ou déposer de la matière et ainsi altérer le terrain en modifiant la structure des couches de matériaux le composant.

Lorsque l'on procède à un dépôt de matière, nous modifions chacune de couches situées dans la partie de l'empreinte définie par le tampon. Si de la matière est présente dans notre texture de stockage, alors nous retirons de la matière de cette texture pour l'ajouter aux couches. Un paramètre représentant l'adhérence de l'objet est pris en compte pour pouvoir déterminer la quantité de matière qui sera alors déposée lors du contact avec le terrain. A l'inverse, pour le retrait, la matière est prise depuis les couches pour être transférée dans la texture de stockage. Le paramètre d'adhérence nous permet ici aussi de spécifier la quantité de matière transitant depuis les couches du terrain vers la texture de stockage.

Ce processus de dépôts et de retrait est répété tout le long de la trajectoire afin d'effectuer le transport de la matière. Comme nous ne faisons que de petites modifications à la surface du sol, les couches de matériaux ne subissent pas de grosses modifications au niveau de leur taille contrairement au processus d'enfoncement qui altère profondément la structure du sol. Les dépôts étant effectués tout au long de la trajectoire, les textures de stockage vont peu à peu se vider et estomper les marques de matières laissées sur le sol.

Afin de pouvoir utiliser le résultat de notre génération et avoir la possibilité de faire un rendu dans n'importe quel logiciel 3D nous exportons plusieurs données. Tout d'abord, le maillage du terrain altéré est enregistré. Ce maillage, composé de triangles, nous laisse voir les modifications géométriques des empreintes sur le sol. Il comportera les abaissements et élévations effectués lors du traitement du terrain. Le deuxième élément que nous utiliserons est une texture permettant de connaître les endroits où l'on trouve différents matériaux. Les matériaux associés seront représentés sous forme de couleurs. Cette texture est calculée au moyen de la topologie de notre modèle. Comme pour la recherche de voisins, chaque empreinte est délimitée puis imprimée sur la texture pour en garder la trace. Cette texture permettra de délimiter les zones où des modifications ont eu lieu lors de la création des empreintes et du transfert de matière le long de la trajectoire. Nous pouvons voir sur la Figure 3.20 une telle carte d'opacité. Le marcheur a emporté ici pratiquement toute la matière présente sous ses pas, et le dépôt qui a suivi est maximal. Nous pouvons de plus observer qu'aucun dépôt n'est effectué avant que le marcheur ne passe dans un matériau pouvant adhérer à ses semelles.

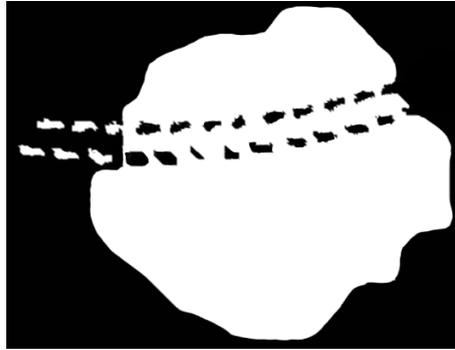


FIGURE 3.20 – La texture générée après le passage d'un marcheur dans une flaque de boue. Nous pouvons observer en blanc les zones où il y a de la matière, c'est à dire la flaque ainsi que les dépôts de matière le long de la trajectoire après la flaque. Les zones noires représentent le matériau de base du terrain qui peut être de la roche dans notre cas.

3.6 Résultats

Nous pouvons voir sur la Figure 3.21 le résultat de l'impression de deux empreintes simples. Sur la gauche, le matériau utilisé est du sable et le poids du sujet est élevé pour y simuler l'enfoncement de son pied. Nous pouvons observer sur cette génération que les bords l'empreinte ont été extrudés pour simuler l'enfoncement franc dans un sable humide. A droite, de la boue a été choisie, nous pouvons constater que le sujet est moins enfoncé dans ce type de matériaux et que les bords ne sont pas extrudés. Cette génération permet de mettre en avant le fait que le matériau choisi ici se comporte d'une façon relativement élastique.



FIGURE 3.21 – Des empreintes de pieds dans un sable de bord de mer en haut et dans de la boue en bas.

La Figure 3.22 montre une trajectoire dans le sable. Cette trajectoire simule un marcheur dans le sable. Nous pouvons observer sur la Figure 3.22 le même marcheur ayant effectué un virage dans le sable. Ces deux simulations ont été effectuées en optant pour des rebords extrudés et ainsi faire apparaître un bord franc et bien découpé pour les empreintes. Nous pouvons voir que la forme de



FIGURE 3.22 – A gauche, une trajectoire de marques de pieds dans le sable. A droite, Un marcheur dans le sable vient de faire demi-tour.

l’empreinte au niveau de la plante du pied est aussi déformée pour épouser la forme de l’empreinte fournie par l’utilisateur.



FIGURE 3.23 – Des centaines d’empreintes sur le sol, nous pouvons voir trois trajectoires se croisant avec un grand nombre d’empreintes sur chacune d’elle. Un tel résultat ne peut être obtenu sans traitement automatique qu’avec beaucoup de temps.

Grâce à notre méthode et au niveau de détails générés, nous pouvons créer un très grand nombre d’empreintes dans un sol. La Figure 3.23 montre des trajectoires comportant des centaines d’empreintes dans un sol désertique. Ces empreintes nous paraissent réalistes bien qu’il y en ai un grand nombre et qu’elles soient vues depuis une grande distance. Cette Figure permet de voir qu’un grand nombre d’estampages peut être effectué sur le terrain par notre modèle alors qu’une méthode classique ne pourrait pas effectuer ce traitement automatiquement.

La Figure 3.24 montre le résultat de la simulation du passage de mouettes sur un sable de littoral. Des agrandissements sont effectués pour montrer l’ajout d’une élévation au niveau des bords de l’empreinte. Cette élévation est présente sur les empreintes de gauche. Elle simule le passage dans du sable sec, dont la matière est transférée sur les rebords de l’empreinte lors de l’enfoncement. Les empreintes qui n’ont pas subi l’élévation de leurs bords simulent des pas dans un sable humide, où les marques sont franches.

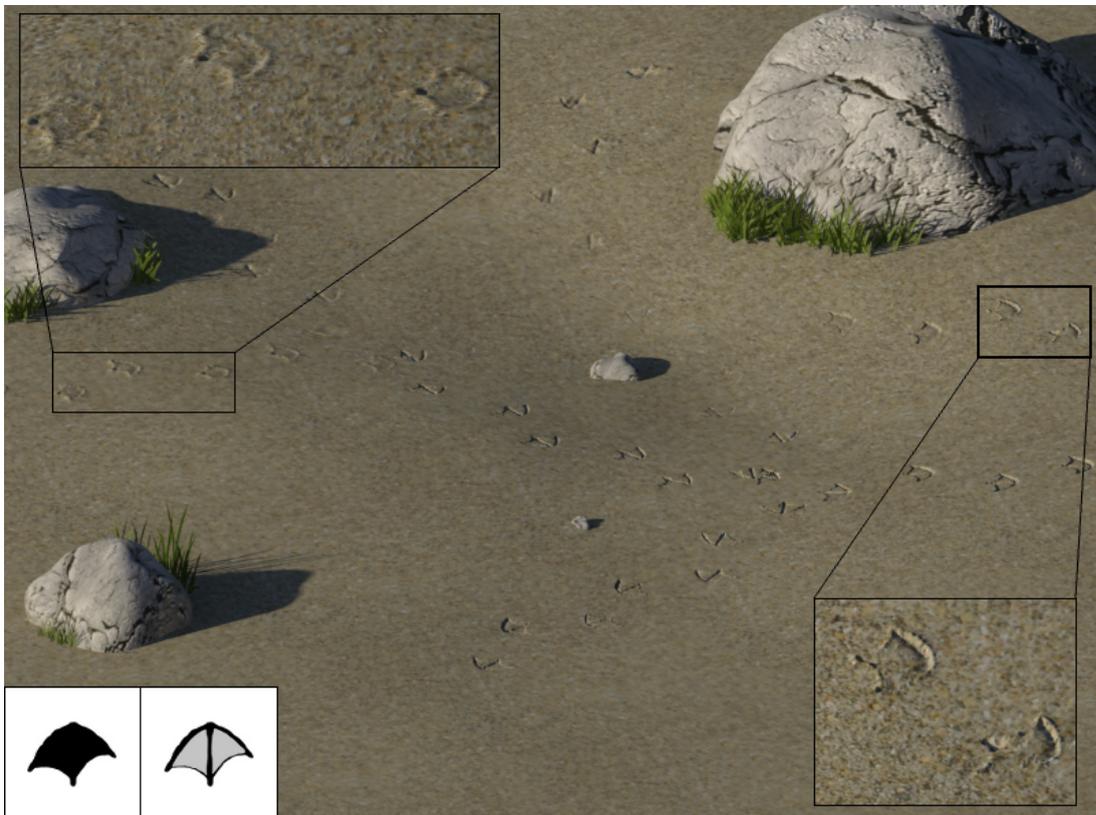


FIGURE 3.24 – Nous pouvons observer sur cette figure plusieurs empreintes de mouettes suivant plusieurs trajectoires. En bas à gauche figurent l’empreinte utilisée ainsi que la carte de hauteur correspondante. Des zooms sont effectués sur des trajectoires différentes pour montrer le résultat de leur génération.



FIGURE 3.25 – A gauche : de la matière déposée par un véhicule. A Droite : un marcheur dépose du sable sur le bitume. Nous pouvons voir sur chaque image les formes d’empreintes utilisées avec en haut la forme générale de l’empreinte et en bas la carte de hauteur correspondante.

Nous pouvons observer sur la Figure 3.25 deux images illustrant le transfert de matière. Nous voyons sur l’image en haut un véhicule ayant transporté de la matière sous ces roues depuis le tas situé à l’arrière-plan jusqu’au sol indéformable au premier plan. Ce transfert de matière s’effectue en plus de la création d’em-



FIGURE 3.26 – Nous pouvons voir sur cette figure les empreintes laissées par un cheval venant de traverser une route.

preintes. En bas nous pouvons observer le même type de dépôts en forçant le trait. Un marcheur a transporté du sable sous ses pieds pour le déposer plus loin sur le bitume. Ces deux images montrent les différentes formes de tampons utilisés qui reflètent directement la forme des dépôts. Nous pouvons voir sur la figure 3.26 les dépôts de matière laissés par un cheval sur du bitume. Différents tampons sont utilisés pour générer les traces laissées au sol par les dépôts de matière.

La Figure 3.27 présente une trajectoire d'empreintes laissées par un cheval au galop sur un chemin forestier. Ces empreintes sont générées en utilisant un patron regroupant plusieurs empreintes disposées de façon à refléter l'allure voulue pour le cheval, ici le galop.

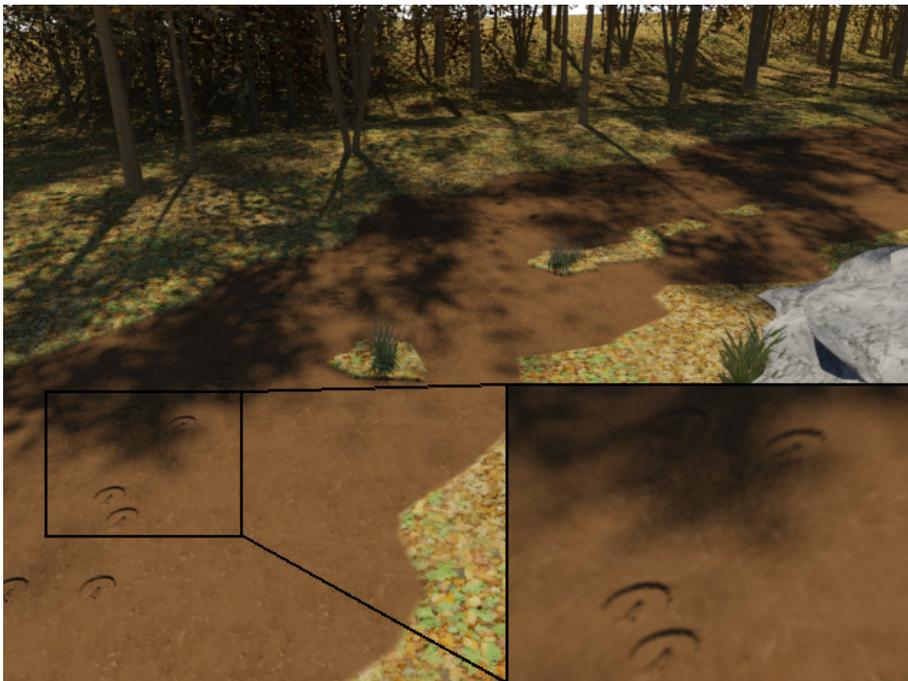


FIGURE 3.27 – Nous pouvons voir sur cette figure les empreintes laissées par un cheval au galop sur un chemin terreux de campagne.

Notre modèle est aussi bien adapté pour simuler les traces laissées par des

véhicules. La figure 3.28 nous montre des empreintes et dépôts laissés par des pneus. Nous pouvons constater que la continuité de la trace est respectée. En effet les réglages de la trajectoire sont définis afin que la fin d'un pas coïncide avec le pas suivant. De ce fait, la simulation d'un pneu roulant sur le sol se fait de façon réaliste, sans discontinuités.



FIGURE 3.28 – Simulation d'une trajectoire de véhicule.

La Figure 3.29 synthétise les deux aspects de notre modèle, à savoir les empreintes dans un sol meuble et les dépôts laissés sur les surfaces. Nous pouvons voir ici une grande quantité de dépôts laissés par plusieurs trajectoires d'engins. L'asphalte est complètement sali de terre entraînée par les roues des engins à partir des côtés de la route. Ces dépôts, bien que régulièrement placés sur leurs trajectoires respectives, conservent un aspect réaliste. En effet, grâce aux tampons et à la gestion de la quantité de matière, la terre est dispersée sur la trajectoire de façon non uniforme. Nous pouvons observer que la quantité de matière déposée est beaucoup plus importante à l'arrivée des engins sur la route que vers la fin de celle-ci.

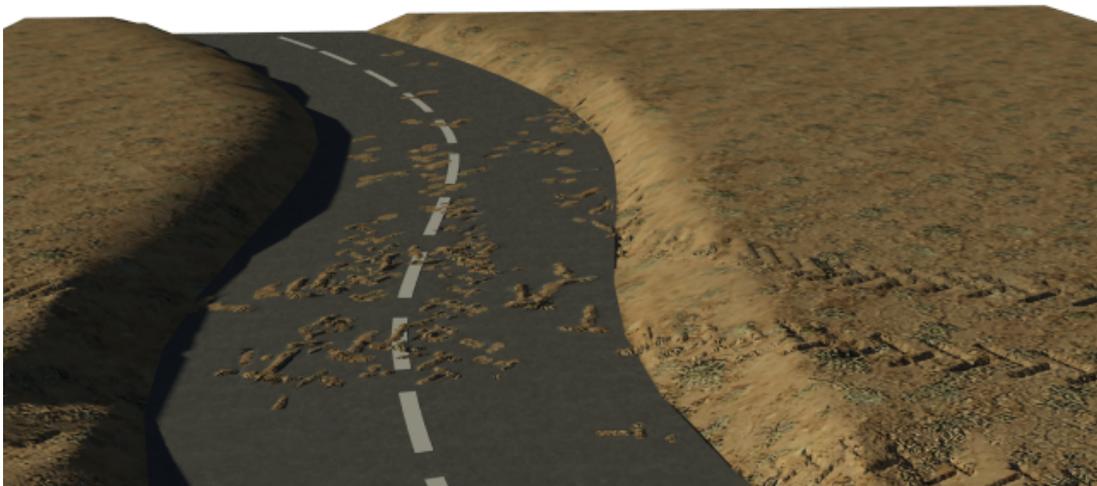


FIGURE 3.29 – Passage d'engins sur une route. Ces engins entrent sur la route par différents endroits et y laissent des marques conséquentes.

3.6.1 Extensions

Pour notre modèle d'altération de terrains, la méthode peut être élargie au vieillissement d'écorces. En effet, bien que semblant assez éloignés, ces deux processus peuvent être effectués par notre modèle. Pour générer des écorces vieilles, notre modèle nécessite des ajouts et des modifications au niveau de ses méthodes.

La principale différence est l'absence de trajectoire. En effet l'écorce de l'arbre est présente sur la totalité de la surface (sans tenir compte des blessures éventuelles). La texture utilisée n'est plus une carte de hauteurs, mais une texture en noir et blanc. Nous générons cette texture en nous appuyant sur les travaux publiés dans [DMG02]. Le blanc symbolise les zones creuses de l'écorce et le noir les zones de surface. Nous pouvons observer sur la figure 3.30 une écorce et sa texture noir et blanc correspondante.

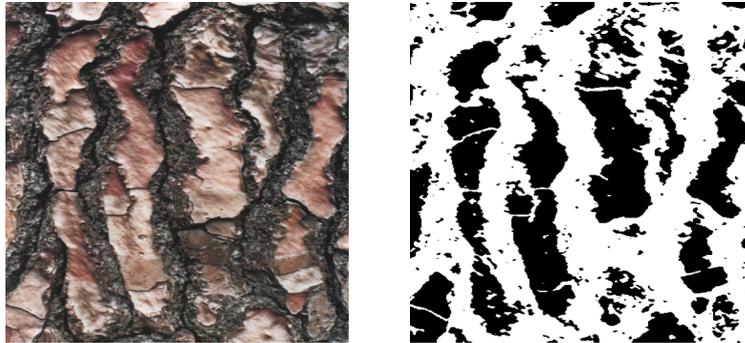


FIGURE 3.30 – A gauche une photo de l'écorce d'un arbre. A droite la texture représentant les zones de surface (noires) et les creux (blancs).

Cette texture est ensuite utilisée un peu comme une empreinte pour être appliquée au maillage par raffinements successifs. Notre méthode s'applique sur un "patch" d'écorce et non sur toute la surface d'un arbre. Il nous suffit par la suite de rendre la texture d'écorce répétable pour obtenir un maillage lui aussi répétable. La Figure 3.31 montre un maillage simple et uniforme triangulaire transformé par notre méthode en patch d'écorce. Grâce à la topologie de notre modèle, nous pouvons rapidement détecter les deux niveaux de zones (surfaces et creux) sur le maillage ainsi généré. De plus, nous pouvons utiliser de nombreux types d'écorces allant de la plus simple à la plus rugueuse comme le montre la figure 3.32.

Une fois ces maillages créés, nos perspectives sont de modifier les hauteurs des brins suivant la hauteur récupérée à partir de la texture d'écorce. Enfin, nous pourrions altérer ce maillage pour en effectuer le vieillissement et l'appliquer à un arbre entier.

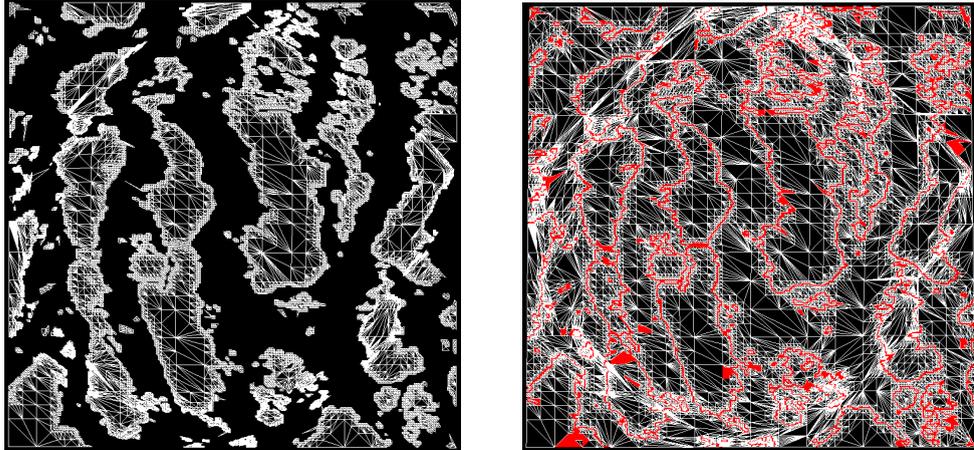


FIGURE 3.31 – Nous pouvons observer sur la gauche le maillage avec les zones de surface de l'écorce et à gauche la détection des bords de ces zones

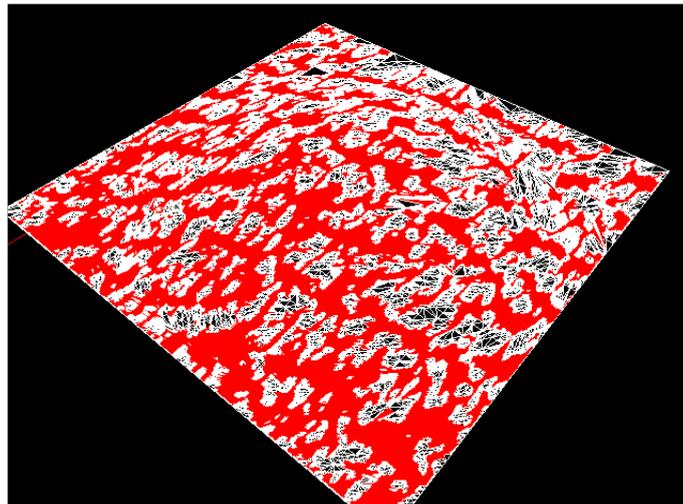


FIGURE 3.32 – Exemple de génération du maillage d'une écorce très rugueuse.

3.7 Conclusion

Notre modèle vise à fournir une méthode efficace pour générer des trajectoires d'empreintes sur le sol. Cette méthode, basée sur l'impression topologique et le transport de matière, offre un moyen rapide et facile de gérer différents types d'empreintes sans recourir à un modèle physique. De plus, les trajectoires établies sont facilement paramétrables en utilisant les données fournies au modèle, à savoir, la forme des empreintes et la trajectoire suivie. La possibilité de gérer des niveaux de détails dépendants de la caméra nous permet de générer des trajectoires comportant un très grand nombre d'empreintes.

Enfin, la méthode de transport de matière permet de simuler les transferts de matières entre les empreintes d'une même trajectoire. Ce transport laissera des traces de passage lorsque le sujet effectuant des empreintes sur le sol marchera dans un sol meuble. Le terrain ainsi modifié sera ensuite exporté sous forme d'un maillage et d'une texture d'opacité pour pouvoir utiliser n'importe quel moteur de rendu.

Grâce à cette méthode, une scène naturelle comportant des terrains pourra facilement se trouver altérée avec des trajectoires d'empreintes. Avec l'importation d'un maillage et la gestion des empreintes selon leur forme au moyen de deux textures, notre modèle permet de générer n'importe quel type d'empreintes sur un sol meuble ou non. De plus, le transfert de matière le long de la trajectoire est géré en prenant en compte la quantité de matière prélevée pour la restituer au terrain.

Conclusion et perspectives

Tout au long de cette thèse, nous nous sommes employés à fournir des outils pour générer des détails dans les scènes naturelles à différents âges. Devant l'immense variété des situations réelles, nous avons choisi d'investiguer deux échelles différentes pour générer des détails et tester nos modèles. Que ce soit pour les feuilles d'arbres ou pour les traces et empreintes sur le sol, les détails peuvent être générés en grande quantité pour peupler la scène naturelle choisie. Les paramètres contrôlant la génération permettent de créer n'importe quelle espèce de feuille en grande quantité ainsi que des empreintes et dépôts de matières sur de longues distances.

Travaux réalisés

Génération de grandes quantités de feuilles à différents âges :

Nous avons développé tout d'abord un générateur de feuilles d'arbres basé sur l'utilisation d'une nouvelle catégorie de L-Système : les 2-G-Cartes L-Systèmes paramétriques. Ces L-Systèmes reposent sur une subdivision topologique en deux dimensions de l'espace. La grammaire fournie par l'utilisateur permettra de générer les nervures de la feuille avec une structure arborescente. Cette structure servira de base pour le corps de la feuille qui sera généré automatiquement.

Après avoir constaté l'importance de la variété des détails dans les scènes naturelles réalistes, nous avons cherché à proposer dans notre modèle la possibilité d'effectuer une telle génération. Nous pouvons obtenir à partir d'une grammaire de base une famille complète de feuille. Cette génération est flexible et tient compte des différences qui peuvent survenir entre chaque membre d'une même espèce.

Nous avons ensuite étendu notre modèle pour pouvoir générer chaque feuille à un âge différent. Ces différents âges induisent un changement au niveau de leurs formes ainsi que de leurs couleurs. Cette diversité de formes et de couleurs au sein de la même espèce est primordiale pour le réalisme des scènes naturelles comportant des feuillages. En effet cet aspect donne un désordre naturel découlant de tous les phénomènes extérieurs affectant différemment chaque processus de croissance sur les feuilles d'un même arbre.

Génération d'empreintes et de traces sur des terrains :

La deuxième partie de nos travaux porte sur la génération d'empreintes et de traces sur le sol. Notre modèle permet, à partir d'une trajectoire et de la forme des empreintes fournies par l'utilisateur, de générer les traces réalisées par un sujet se déplaçant sur le sol. Nous avons mis en place des mécanismes génériques pour la création d'empreintes et de dépôts de matière sur un sol.

Pour effectuer de telles traces, nous avons utilisé un modèle topologique couplant le modèle des 2-G-Cartes avec des couches de matériaux. Pour une trajectoire donnée, les différents points d'impact sont calculés puis imprimés au moyen de raffinements topologiques successifs sur le maillage. Une fois le niveau de détail désiré atteint, les empreintes sont enfoncées en suivant la forme voulue ainsi que les paramètres inhérents au sujet comme son poids ou bien sa vitesse. Les enfoncements ainsi effectués peuvent être paramétrés suivant différents aspects par l'utilisateur pour rester au plus proche de la simulation du matériau où les empreintes sont réalisées.

La connaissance de la structure du sol nous permet de rapidement stocker la matière enlevée lors des enfoncements pour la réutiliser en aval. Cette étape est assurée par la gestion de la quantité de matière retirée et déposée sur le modèle tout au long de la trajectoire. Cette quantité de matière est stockée dans une texture qui est mise à jour à chaque génération d'empreinte sur le sol puis utilisée sous forme de tampon pour être déposée sur la surface du sol et pour gérer une quantité de matière cohérente le long de la trajectoire.

Perspectives

Pour la suite de nos travaux, nous prévoyons d'étendre la modélisation de détails dans les scènes naturelles à d'autres éléments. En effet, en utilisant une grammaire simple et en ajoutant la possibilité d'utiliser les 3-G-Cartes, il nous sera possible de générer toutes sortes de familles de détails et d'en peupler des scènes naturelles.

Dans ce but nous envisageons d'améliorer la conception de la grammaire de création des détails. Cette création se fait pour l'instant uniquement en insérant sciemment des règles de production. Nous envisageons de pouvoir déterminer ces règles de façon interactive et automatique. L'exportation des détails autre que des feuilles nécessitera de revoir la méthode d'export et de vieillissement. Deux textures et un masque de mélange ne seront plus suffisants pour décrire le vieillissement colorimétrique de certains détails. Ces travaux permettront d'explorer plus complètement les détails à une petite échelle.

A une échelle supérieure, nous avons aussi comme projet de continuer nos travaux sur la génération d'empreintes et de traces sur le sol. Ces travaux seront

tournés vers le déplacement de matériaux grâce à notre modèle topologique. En effet l'ajout de plusieurs effets comme une adhérence sélective en fonction du type et de la forme des surfaces entrant en contact avec le sol ou bien la génération d'éclaboussures pour les matériaux relativement fluides seraient des améliorations intéressantes de notre méthode.

Enfin, nos travaux nous poussent vers une approche plus générale sur la génération de détails. Une méthode unifiée de génération de détails pour les scènes naturelles réalistes devrait-elle forcément consister en une collection de méthodes de génération pour chaque famille de détail ? Notre approche sera plutôt d'utiliser la même méthode de génération permettant d'assembler et de disperser ces détails dans une scène naturelle. Une grammaire différente pourra être utilisée pour chaque détail puis une grammaire plus générale définie pour disposer ces détails dans la scène voulue.

Annexes

Grammaire Annexe .1. Feuille elliptique :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 20, 0, 0, 0, 20, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 5, 0, 0, 0, 5, 1)
#define F(4, 5, 0, 0, 0, 5, 1)
#define anglesecondaire = 70
#define c = 5
#axiome : A

p00 A → A[B(, , , , ,)]E
p00 B → B[D(, , , , ,)]E
p00 D → D[B(, , , , ,)]E
p01 D → D[E(, , , < anglesecondaire > , ,)]C1
p01 D → D[F(, , , < -anglesecondaire > , ,)]C2

p02 E → E[E(< c + etape > , , < -c > , < c + etape > ,)]E
p02 F → F[F(< c + etape > , , < c > , < c + etape > ,)]E

```

Grammaire Annexe .2. Feuille lancéolée :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 40, 0, 0, 0, 40, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 10, 0, 0, 0, 10, 1)
#define F(4, 10, 0, 0, 0, 10, 1)
#define anglesecondaire = 70
#define c = 5
#axiome : A

p00 A → A[B(, , , , ,)]E
p00 B → B[D(, , , , ,)]E
p00 D → D[B(, , , , ,)]E
p01 D → D[E(, , , < anglesecondaire >, ,)]C1
p01 D → D[F(, , , < -anglesecondaire >, ,)]C2
p02 E → E[E(, < 1 + etape >, , < -c >, < 1 + etape >,)]E
p02 F → F[F(, < 1 + etape >, , < c >, < 1 + etape >,)]E

```

Grammaire Annexe .3. Feuille linéaire :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 80, 0, 0, 0, 80, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 10, 0, 0, 0, 10, 1)
#define F(4, 10, 0, 0, 0, 10, 1)
#define anglesecondaire = 45
#axiome : A

p00 A → A[B(, , , , ,)]E
p00 B{{etape == 1}} → B[D(, , , , ,)]E
p00 D{{etape == 1}} → D[B(, , , , ,)]E
p00 B{{etape > 1}} → B[B(, , , , ,)]E
p01 D → D[E(, , , < anglesecondaire >, ,)]C1
p01 D → D[F(, , , < -anglesecondaire >, ,)]C2

p02 E{{etape <= (etapeFinale/3)}} → E[E(, 30, , , -15, 30,)]E
p02 F{{etape <= (etapeFinale/3)}} → F[F(, 30, , , 15, 30,)]E
p02 E{{etape > (etapeFinale/3)}} → E[E(, 80, , , 80,)]E
p02 F{{etape > (etapeFinale/3)}} → F[F(, 80, , , 80,)]E

```

Grammaire Annexe .4. Feuille ovale :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 20, 0, 0, 0, 20, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 10, 0, 0, 0, 10, 1)
#define F(4, 10, 0, 0, 0, 10, 1)
#define anglesecondaire = 90
#axiome : A

p00 A → A[B(, , , , ,)]E
p00 B → B[D(, , , , ,)]E
p00 D → D[B(, 10, , , , 10,)]E
p01 D → D[E(, , , , < anglesecondaire - (etape * 8) >, ,)]C1
p01 D → D[F(, , , , < -anglesecondaire + (etape * 8) >, ,)]C2
p02 E → E[E(, < 7 + etape >, , , -8, < 7 + etape >,)]E
p02 F → F[F(, < 7 + etape >, , , 8, < 7 + etape >,)]E
p03 B{}{etape == etapeFinale}{} → B[B(, , , , ,)]E

```

Grammaire Annexe .5. Feuille oblongue :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 40, 0, 0, 0, 40, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 5, 0, 0, 0, 5, 1)
#define F(4, 5, 0, 0, 0, 5, 1)
#define secondaire = 70
#axiome : A

p00 A → A[B(, , , , ,)]E
p00 B → B[D(, , , , ,)]E
p00 D → D[B(, , , , ,)]E
p01 D{}{etape > etapeFinale/2}{} → D[E(, , , , < secondaire >, ,)]C1
p01 D{}{etape > etapeFinale/2}{} → D[F(, , , , < -secondaire >, ,)]C2
p02 E → E[E(, < etape * 2 >, , , -5, < etape * 2 >,)]E
p02 F → F[F(, < etape * 2 >, , , 5, < etape * 2 >,)]E

```

Grammaire Annexe .6. Feuille obovale :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 20, 0, 0, 0, 20, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 10, 0, 0, 0, 10, 1)
#define F(4, 10, 0, 0, 0, 10, 1)
#define anglesecontaire = 100
#axiome : A

p00 A → A[B(, , , , ,)]E
p00 B → B[D(, , , , ,)]E
p00 D → D[B(, 10, , , , 10,)]E
p01 D → D[E(, , , , < anglesecontaire - etape * 3 >, ,)]C1
p01 D → D[F(, , , , < -anglesecontaire + etape * 3 >, ,)]C2
p02 E → E[E(, < etape + 1 >, , , -4, < etape + 1 >,)]E
p02 F → F[F(, < etape + 1 >, , , 4, < etape + 1 >,)]E
p00 B{{etape == etapeFinale}} → B[B(, , , , ,)]E

```

Grammaire Annexe .7. Feuille obtuse :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 20, 0, 0, 0, 20, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 5, 0, 0, 0, 5, 1)
#define F(4, 5, 0, 0, 0, 5, 1)
#define anglesecontaire = 70
#axiome : A

p00 A → A[B(, , , , ,)]E
p00 B → B[D(, , , , ,)]E
p00 D → D[B(, , , , ,)]E
p01 D → D[E(, , , , < anglesecontaire >, ,)]C1
p01 D → D[F(, , , , < -anglesecontaire >, ,)]C2
p02 E → E[E(, < etape * 3 >, , , -5, < etape * 3 >,)]E
p02 F → F[F(, < etape * 3 >, , , 5, < etape * 3 >,)]E

```

Grammaire Annexe .8. Feuille oblancéolée :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 20, 0, 0, 0, 20, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 5, 0, 0, 0, 5, 1)
#define F(4, 5, 0, 0, 0, 5, 1)
#define anglesecondaire = 70
#axiome : A

p00 A → A[B(, , , , ,)]E
p00 B → B[D(, , , , ,)]E
p00 D → D[B(, , , , ,)]E
p01 D → D[E(, , , < anglesecondaire > , ,)]C1
p01 D → D[F(, , , < -anglesecondaire > , ,)]C2
p02 E → E[E(< etape > , , -5, < etape > ,)]E
p02 F → F[F(< etape > , , 5, < etape > ,)]E

```

Grammaire Annexe .9. Feuille Falciforme :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 80, 0, 0, 0, 80, 1)
#define D(4, 10, 0, 0, 0, 10, 1)
#define E(4, 10, 0, 0, 0, 10, 1)
#define F(4, 10, 0, 0, 0, 10, 1)
#define anglesecondaire = 45
#axiome : A

p00 A- > A[B(, , , , ,)]E
p00 B{ }{ etape == 1 }{ } → B[D(, , , , ,)]E
p00 D{ }{ etape == 1 }{ } → D[B(, , , , ,)]E
p00 B{ }{ etape > 1 }{ } → B[B(, , , < -etape - 1, 8 > , ,)]E
p01 D → D[E(, , , < anglesecondaire > , ,)]C1
p01 D → D[F(, , , < -anglesecondaire > , ,)]C2
p02 E{ }{ etape <= (etapeFinale/3) }{ } → E[E(, 30, , -15, 30, )]E
p02 F{ }{ etape <= (etapeFinale/3) }{ } → F[F(, 30, , 15, 30, )]E
p02 E{ }{ etape > (etapeFinale/3) }{ } → E[E(, 80, , < -etape > , 80, )]E
p02 F{ }{ etape > (etapeFinale/3) }{ } → F[F(, 80, , < -etape > , 80, )]E

```

Grammaire Annexe .10. Feuille dentée :

```

#define A(4, 1, 0, 0, -90, 1, 0.1)
#define B(4, 3, 0, 0, 0, 3, 0.1)
#define C(4, 1, 0, 0, 0, 2, 0.1)
#define D(4, 1, 0, 0, 0, 0.2, 0.1)
#define E(4, 1, 0, 0, 0, 1, 0.1)
#define F(4, 1, 0, 0, 0, 1, 0.1)
#define G(4, 1, 0, 0, 0, 0.1, 0.1)
#define H(4, 1, 0, 0, 0, 0.1, 0.1)
#define angle = 60
#define remplissage = 20
#axiome : A

p00 A → A[B(, , 2, , ,)]E
p00 B → B[D(, , 2, , ,)]E
p00 D → D[B(, , 2, , ,)]E
p01 D → D[E(, , , < math.random(60, 60) >, ,)]C1
p01 D → D[F(, , , < -math.random(60, 60) >, ,)]C2
p02 E → E[E(, , , < -math.random(0, 5) - 1 >, ,)]E
p02 F → F[F(, , , < math.random(0, 5) + 1 >, ,)]E
p03 B{{etape == etapeFinale}} → B[B(, , , , ,)]E
p03 E{{etape == etapeFinale}} → E[G(, , , , ,)]E
p03 F{{etape == etapeFinale}} → F[H(, , , , ,)]E
p03 G{{etape == etapeFinale}} → G[F(, , , < math.random(10, 35) >
, 0.3, 0.01)]C1
p03 H{{etape == etapeFinale}} → H[E(, , , < -math.random(10, 35) >
, 0.3, 0.01)]C2

```

Grammaire Annexe .11. Feuille trifoliée :

```

#define A(4, 10, 0, 0, -90, 10, 1)
#define B(4, 1, 0, 0, 0, 1, 1)
#define C(4, 10, 0, 0, 0, 10, 1)
#define D(4, 10, 0, 0, 0, 1, 1)
#define E(4, 10, 0, 0, 0, 10, 1)
#define F(4, 10, 0, 0, 0, 10, 1)
#define G(7, 10, 0, 0, 0, 10, 1)
#define angle = 5
#define remplissage = 20
#define ordreremplissage = 1
#define l = 8
#axiome : A

p00 A → A[G]E
p00 G → G[B(, , , , 120, , )]C1
p00 G → G[B(, , , , , )]C3
p00 G → G[B(, , , , -120, , )]C5
p01 B → B[D(, , , , , )]E
p01 D → D[B(, < 10/(etape + 10) >, , , , < 10/(etape + 10) >, )]E
p02 D → D[E(, , , , < 60 - etape/2 >, , )]C1
p02 D → D[F(, , , , < -60 + etape/2 >, , )]C2
p03 E → E[E(, < l >, , , < -angle >, < l >, )]E
p03 F → F[F(, < l >, , , < angle >, < l >, )]E
p04 B{}{etape == etapeFinale}{} → B[B(, 5, , , , 5, )]E

```

Bibliographie

- [ABZO06] A.S. Aquilio, J.C. Brooks, Y. Zhu, and G.S. Owen. Real-time gpu-based simulation of dynamic terrain. *ISVC06*, pages 891–900, 2006.
- [Bas06] S.K. Basu. Modeling and dynamics of plant leaves. *Dept. of Computer Science, UNC Chapel Hill*, 2006.
- [BDE04] M. Braitmaier, J. Diepstraten, and T. Ertl. Real-time rendering of seasonal influenced trees. In *Proceedings of Theory and Practice of Computer Graphics 2004*, pages 152–159. Eurographics UK, 2004.
- [Bek69] M. Bekker. *Introduction to Terrain-Vehicle Systems*. University of Michigan Press, 1969.
- [BF01] B. Benes and R. Forsbach. Layered data representation for visual simulation of terrain erosion. In *SCCG '01 : Proceedings of the 17th Spring conference on Computer graphics*, page 80, Washington, DC, USA, 2001. IEEE Computer Society.
- [BF02] B. Benes and R. Forsbach. Visual simulation of hydraulic erosion. In *WSCG*, pages 79–94, 2002.
- [BLD⁺06] Laurent Bousquet, Thomas Lavergne, Thierry Deroin, Jean-Luc Widlowski, Ismaël Moya, and Stéphane Jacquemoud. Multispectral and multiangular measurement and modeling of leaf reflectance and transmittance. *Quantitative Remote Sensing*, 2006.
- [BR97] Gladimir V. G. Baranoski and Jon G. Rokne. An algorithmic reflectance and transmittance model for plant tissue. *Computer Graphics Forum*, 16 :141–150, 1997.
- [BR01] Gladimir V. G. Baranoski and Jon G. Rokne. Efficiently simulating scattering of light by leaves. In *The Visual Computer*, volume 17, pages 491–505, 2001.
- [CL92] X. Chen and P. Lienhardt. Modelling and programming evolutions of surfaces. 1992. *Computer Graphics Forum*, Vol. 2, no. 5.
- [CLH96] Benoit Chanclou, Annie Luciani, and Arash Habibi. Physical models of loose soils dynamically marked by a moving object. In *CA '96 : Proceedings of the Computer Animation*, page 27, Washington, DC, USA, 1996. IEEE Computer Society.

- [CLS07] Xingquan Cai, Jinhong Li, and Zhitong Su. *Real-Time Visualization of Dynamic Terrain in Off-Road Driving Simulation System*. IEEE Computer Society, Washington, DC, USA, 2007.
- [CMF98] N Chiba, K Muraoka, and K. Fujita. An erosion model based on velocity fields for the visual simulation of mountains scenery. *Journal of Visualization and Computer Animation*, 9(4) :185–194, 1998.
- [CZXY07] Guojun Chen, Jing Zhang, Xiaoli Xu, and Yuan Yin. Real-time visualization of tire tracks in dynamic terrain with lod. *Edutainment*, 4469 :655–666, 2007.
- [DMG02] Jean-Michel Dischler, Karl Maritaud, and Djamchid Ghazanfarpour. Coherent bump map recovery from a single texture image. In *Graphics Interface*, pages 201–208, 2002.
- [EL92] H. Elter and P. Lienhardt. Extension of the notion of map for the representation of the topology of cellular complexes. In *4th Canadian Conference on Computational Geometry*, St John’s, Canada, 1992.
- [Fed02] P. Federl. *Modeling Fracture Formation on Growing surfaces*. PhD thesis, University of Calgary, 2002.
- [GHDS03] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. Discrete shells. In *SCA ’03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 62–67, Aire-la-Ville, Switzerland, 2003. Eurographics Association.
- [GS85] Leonidas Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of voronoi. *ACM Trans. Graph.*, 4(2) :74–123, 1985.
- [HTK00] T Hirota, Y. Tanoue, and T Kaneko. An empirical stochastic model for geometry of two-dimensional crack growth in soil. *Geoderma*, 96 :263–276, 2000.
- [KCB07] Pierre KRAEMER, David CAZIER, and Dominique BECHMANN. Multiresolution half-edges. In *Proceedings of SCCG’07 (Spring Conference on Computer Graphics)*. ACM Press, april 2007.
- [KCB09] Pierre KRAEMER, David CAZIER, and Dominique BECHMANN. Extension of half-edges for the representation of multiresolution subdivision surfaces. *The Visual Computer*, 25(2) :none, 2009.
- [Lam87] David Lam. Tokamak physics engine. An open source, real-time physics engine, 1987.
- [Lie88] P. Lienhardt. Free-form surfaces modeling by evolution simulation. Septembre 1988. Eurographics’88, Nice, France, pp. 327-341.
- [Lie89] P. Lienhardt. 2-G-maps : a model for the manipulation of general 2-dimensional subdivisions. In *Proc. of International Conference on Computer-Aided Design and Computer Graphics*, 1989.

- [Lie91] P. Lienhardt. Topological models for boundary representation : a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23, 1991.
- [Lie94a] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal on Computational Geometry and Applications*, 4(3) :275–324, 1994.
- [Lie94b] Pascal Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. 1994. *International Journal on Computational Geometry and Applications*, Vol. 4, n° 3, pp. 275-324.
- [LM93] Xin Li and J. Michael Moshell. Modeling soil : Realtime dynamic models for soil slippage and manipulation. In *Computer Graphics Proceedings, Annual Conference Series*, pages 361–368, 1993.
- [LTBG09] C. Lam, O. Terraz, N. Benmounah, and D. Ghazanfarpour. Geometrical aspects of wood aging based on 3gmap l-system growth model. *Proceedings of Computer Graphics, Visualization, Computer Vision and Image Processing'2009*, pages 249–254, June 2009.
- [MCK⁺01] S. Mochizuki, D. Cai, T. Komiri, H. Kimura, and R. Hori. Virtual autumn coloring system based on biological and fractal model. In *PG '01 : Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, page 348, Washington, DC, USA, 2001. IEEE Computer Society.
- [MHW92] Przemyslaw Prusinkiewicz Mark Hammel and Brian Wyvill. Modelling compound leaves using implicit contours. *Proceedings of Computer Graphics International '92*, pages 119–212, 1992.
- [MKM89] F. Kenton Musgrave, Craig E. Kolb, and Robert S. Mace. The synthesis and rendering of eroded fractal terrains. *Computer Graphics*, 23(3) :41–50, 1989.
- [MMPP03] Lars Mundermann, Peter MacMurchy, Juraj Pivovarov, and Przemyslaw Prusinkiewicz. Modeling lobed leaves. *Computer Graphics International*, pages 60–67, 2003.
- [NCN96] K. Muroaka N. Chiba, K. Ohshida and S. Nobuji. Visual simulation of leaf arrangement and autumn colors. *The Journal of Visualization and Computer Animation*, 7 :79–93, 1996.
- [ON03] Koichi Onoue and Tomoyuki Nishita. Virtual sandbox. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, PG '03, pages 252–, Washington, DC, USA, 2003. IEEE Computer Society.
- [Pey11] Alexandre Peyrat. Rapport sir_2010_11 : Une méthode d'ajout de détails et d'altération des terrains. *Rapport de recherche, Equipe SIR, DMI, Xlim UMR CNRS 6172*, 2011.

- [PGMG09] Adrien Peytavie, Eric Galin, Stephane Merillou, and Jerome Grosjean. Arches : a Framework for Modeling Complex Terrains. *Computer Graphics Forum (Proceedings of Eurographics)*, 28(2) :457–467, 2009.
- [PL96] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, USA, 1996.
- [PMKL01] Przemyslaw Prusinkiewicz, Lars Muendermann, Radoslaw Karwowski, and Brendan Lane. The use of positional information in the modeling of plants. In *Proceedings of SIGGRAPH 2001*, pages 289–300. ACM Press, 2001.
- [PMRDM96] E. Perrier, C. Mullon, M. Rieu, and G. De Marsily. Computer construction of fractal soil structures : simulation of their hydraulic and shrinkage properties. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, 33 :259A–260A(2), September 1996.
- [Pru04] Przemyslaw Prusinkiewicz. Art and science of life : Designing and growing virtual plants with l-systems. *Acta Hort*, 630 :15–28, 2004.
- [PTMG08] Alexandre Peyrat, Olivier Terraz, Stephane Merillou, and Eric Galin. Generating vast varieties of realistic leaves with parametric 2gmap l-systems. *The Visual Computer*, 24(7-9) :807–816, 2008.
- [QTZ⁺06] Long Quan, Ping Tan, Gang Zeng, Lu Yuan, Jingdong Wang, and Sing Bing Kang. Image-based plant modeling. *ACM Trans. Graph.*, 25(3) :599–604, 2006.
- [RFL⁺05] A. Runions, M. Fuhrer, B. Lane, P. Federl, A. Rolland-Lagan, and P. Prusinkiewicz. Modeling and visualization of leaf venation patterns. *Proceedings of SIGGRAPH*, 2005.
- [RLFS02] Yodthong Rodkaew, Chidchanok Lursinsap, Tadahiro Fujimoto, and Suchada Siripant. Modeling leaf shapes using l-systems and genetic algorithms. In *In International Conference NICOGRAPH (April)*, pages 73–78, 2002.
- [SBH99] Robert W. Sumner, James F. O Brien, and Jessica K. Hodgins. Animating sand, mud, and snow. *Comput. Graph. Forum*, 18(1) :17–26, 1999.
- [Skj88] P. Skjeltorp, A. T. and Meakin. Fracture in microsphere monolayers studied by experiment and computer simulation. *Nature*, 1988.
- [TGM⁺08] O. Terraz, G. Guimberteau, S. Merillou, D. Plemenos, and D. Ghazanfarpour. 3gmap l-systems : an application to the modelling of wood. *The Visual Computer*, 25(2) :165–180, 2008.

- [Tut84] W. Tuttle. *Graph theory, Encyclopedia of Mathematics and its Applications*. Addison Wesley, 1984.
- [vOV96] C. W. A. M. van Overveld and Marie Luce Viaud. Sticky splines : definition and manipulation of spline structures with maintained topological relations. *ACM Trans. Graph.*, 15(1) :72–98, 1996.
- [VPLL06] G. Valette, S. Prevost, L. Lucas, and J. Léonard. Soda project : a simulation of soil surface degradation by rainfall. *Computers and Graphics*, 30(4) :494–506, aug 2006.
- [VPLL08] G. Valette, S. Prevost, L. Lucas, and J. Léonard. A dynamic model of cracks development based on a 3d discrete shrinkage volume propagation. *Computer Graphics Forum*, 27(1), 2008.
- [WWD⁺05] Lifeng Wang, Wenle Wang, Julie Dorsey, Xu Yang, Baining Guo, and Heung-Yeung Shum. Real-time rendering of plant leaves. *Siggraph'05*, 24 :712–719, 2005.
- [YRL04] S. Siripant Y. Rodkaew, P. Chongstitvatana and C. Lursinsap. Modeling plant leaves in marble-patterned colours with particle transportation system. In *4th International Workshop on Functional-Structural Plant Models*, pages 391–397, Montpellier, France, 2004. C. Godin et al.
- [ZTT⁺07] Ya-Lun Zeng, Charlie Irawan Tan, Wen-Kai Tai, Mau-Tsuen Yang, Cheng-Chin Chiang, and Chin-Chen Chang. A momentum-based deformation system for granular material. *Comput. Animat. Virtual Worlds*, 18 :289–300, 2007.