

THÈSE

présentée

devant l'**UNIVERSITE CLAUDE BERNARD - LYON 1**

pour l'obtention

du **DIPLOME DE DOCTORAT**

(arrêté du 25 avril 2002)

présentée et soutenue publiquement le

17 novembre 2006

par

M. Brett DESBENOIT

Modélisation et simulation de scènes naturelles complexes

sous la direction de

Samir AKKOUCHE

et

Eric GALIN

COMPOSITION DU JURY

Jean-Michel Dischler	Rapporteur	Professeur, Université de Strasbourg
Yannick Remion	Rapporteur	Professeur, IUT de Reims
Jean-Pierre Jessel	Examineur	Professeur, Université de Toulouse
Stéphane Mérillou	Examineur	Maître de Conférences, Université de Limoges
Samir Akkouche	Directeur de thèse	Professeur, Université Lyon 1
Eric Galin	Co-directeur de thèse	Maître de Conférences, Université Lyon 1

Table des matières

Chapitre 1 Introduction	1
Chapitre 2 Etat de l'art	7
2.1 Processus de génération de scènes naturelles	9
2.1.1 Génération de terrains	9
2.1.2 Modélisation des végétaux	11
2.1.3 Dissémination de végétaux	13
2.2 Modélisation de détails sur les végétaux	16
2.2.1 Modélisation des poils et des épines	16
2.2.2 Création de l'écorce	17
2.2.3 Modélisation des feuilles et des pétales	19
2.3 Vieillissement	21
2.3.1 Techniques de vieillissement de surfaces	21
2.3.2 Fissures et Fractures	24
2.4 Synthèse	30
Chapitre 3 Simulation de la propagation de lichen	33
3.1 Biologie du lichen	35
3.2 Dispersion du lichen	37
3.2.1 Dépôt manuel	38
3.2.2 Dépôt reposant sur l'accessibilité	38
3.3 Algorithme de propagation du lichen : l'Open-DLA	39
3.3.1 Le modèle du DLA	40
3.3.2 Algorithme de DLA intégrant la division cellulaire	41
3.3.3 Marche aléatoire	42
3.3.4 Détection de collision	43
3.3.5 Calcul de la probabilité d'agrégation	44
3.4 Atlas de lichen	47
3.4.1 Processus de substitution	48

3.4.2	Remplissage de l'Atlas	49
3.5	Synthèse	49
Chapitre 4	Simulation de la chute et du vieillissement des feuilles	55
4.1	Phénomènes chimiques et biologiques et coloration des feuilles	58
4.2	Modélisation de l'atlas	59
4.3	Processus de vieillissement	61
4.3.1	Caractéristiques de l'environnement	61
4.3.2	L'Open Markov-Chain	62
4.4	Simulation de la chute de feuilles	64
4.4.1	Processus de détachement	65
4.4.2	Simulation de la chute des feuilles	65
4.4.3	Compactage des tas de feuilles	68
4.5	Résultat et discussion	68
Chapitre 5	Création de fissures réalistes	73
5.1	Modélisation des fissures	76
5.1.1	Modèle de fissures génériques	76
5.1.2	Edition d'un motif de fissure	77
5.2	Ajout d'une fissure	78
5.2.1	Création du squelette	79
5.2.2	Création du volume de la fissure	80
5.2.3	Cas particuliers	80
5.2.4	Résultats	83
5.3	Animation de la propagation d'une fissure	88
5.4	Conclusion	89
Chapitre 6	Conclusion et perspectives	93
Bibliographie		99

Chapitre 1

Introduction

La modélisation de scènes naturelles a fait l'objet de beaucoup d'attention au cours de ces 30 dernières années dans le domaine de l'informatique graphique. L'objectif est de créer des paysages les plus réalistes possibles mais aussi de reproduire des environnements existants. Du fait de l'augmentation rapide des capacités de calcul, les scènes de synthèse que l'on retrouve actuellement dans les jeux vidéos, mais aussi dans l'industrie du cinéma, sont de plus en plus détaillées ce qui les rend de plus en plus crédibles. Le réalisme est également très recherché dans le domaine de la réalité virtuelle, avec notamment, la visite de sites historiques où il participe grandement à l'immersion de l'utilisateur.

Modélisation de scènes naturelles

La génération de scènes naturelles pose deux problèmes fondamentaux : d'une part, la modélisation détaillée des éléments, tant d'un point de vue géométrique que de leur texture et apparence, et d'autre part, leur répartition dans la scène selon l'environnement (éclairage, compétition avec d'autres espèces...).

Les méthodes de modélisation détaillées des objets peuvent être classifiées selon qu'elles sont réalisées en une passe ou en plusieurs.

Dans les méthodes en une passe, l'objet est créé directement à son niveau de détail le plus élevé. Pour cela, le designer dispose généralement d'outils spécifiques. L'objet peut également être généré par l'intermédiaire d'une simulation. C'est le cas par exemple pour la modélisation de l'écorce des arbres et des plantes poilues. Dans une telle approche, les différents niveaux de détail sont obtenus principalement par des méthodes de dégradations. C'est à dire que l'objet est rééchantillonné.

L'approche utilisée par les méthodes en plusieurs passes consiste à créer les objets à un faible niveau de détails puis à augmenter leur réalisme par le biais de simulations ou d'outils spécifiques. Dans la majorité des cas, les méthodes développées sont des méthodes de dégradations. On peut citer parmi les phénomènes les plus étudiés : l'érosion, la corrosion, les patines, le transport et le dépôt de poussières, les mousses et également les fissures. La majorité de ces

méthodes travaillent dans l'espace texture. Cela n'est pas toujours suffisant pour obtenir un effet réaliste dans le cas de vues rapprochées. De plus, dans cette approche, les simulations sont très largement utilisées. Leurs principaux inconvénients sont d'une part un temps de calcul conséquent et d'autre part, elles nécessitent un grand nombre de paramètres parfois peu intuitifs. Enfin, elles présentent également l'inconvénient de n'être n'y prédictibles, n'y reproductibles. Toutefois, ces approches permettent de définir au moins deux niveaux de détail de manière très naturelle : le premier niveau ne contenant que le support et le second le support décoré par les éléments ajoutés.

Le réalisme d'une scène virtuelle est en partie le résultat de l'accumulation d'une grande variété de détails. Leur répartition dans la scène joue un rôle très important car elle fournit des informations visuelles telles que le micro climat, le type de faune et de flore présents. Afin que la scène soit réaliste, il est nécessaire que toutes ces données concordent. Ainsi, la présence d'un carré de pelouse anglaise au milieu d'un désert aride serait visuellement dérangement. A contrario, dans un environnement forestier humide, on recherchera de la mousse sur le coté nord des arbres.

L'un des éléments le plus étudié est la répartition d'espèces végétales dans un écosystème donné. Dans la littérature, on retrouve deux approches classiques. La première approche consiste à partir du niveau le plus haut : l'écosystème. A partir d'une carte de densité, les espèces végétales sont réparties dans l'environnement en fonction de leurs affinités respectives. Cette carte peut être obtenue à partir d'observation, par simulation ou encore produite à la main par un artiste. La deuxième approche souvent utilisée consiste à partir du niveau le plus bas : la plante. Les nouveaux éléments de la flore sont ajoutés itérativement en prenant en compte leur voisinage.

Un très grand nombre de phénomènes naturels sont simulés par ces deux approches. C'est le cas entre autre de la compétition pour le territoire et les ressources (eau, lumière, minéraux. . .) ou encore des échanges symbiotiques et asymbiotiques.

En revanche, la simulation du vieillissement de l'aspect des végétaux a été très peu étudiée. Dans les approches précédentes, lorsqu'une plante meurt, elle est retirée de la simulation. Or, lorsque la plante végète ou qu'elle commence à dépérir, ses feuilles jaunissent et tendent vers le brun avant de tomber. De même, la tige brunit et s'assèche. Tous ces éléments ont un très fort impact visuel sur la scène et leur présence devient alors très importante. Ceci montre les limites du modèle qui consiste à dire que lorsque la plante n'a plus assez de ressources elle disparaît. De plus, les feuilles et les tiges qui pourrissent, enrichissent le sol en matière organique utile aux autres plantes.

L'ajout de tous ces éléments dans la scène, nécessaire au réalisme, pose le problème de la gestion de la mémoire et de leur rendu. Dans une certaine mesure, ces deux éléments sont liés. En effet, les problèmes de gestion de mémoire sont différents selon que l'on souhaite immerger un utilisateur dans un environnement virtuel tel que c'est le cas dans un jeu vidéo, ou alors que l'on souhaite générer une animation telle que l'on peut le voir dans les films d'animation réalisés en images de synthèse.

Dans le cas de la navigation dans une scène virtuelle, l'utilisation des niveaux de détails n'est plus suffisante car le problème ne vient plus uniquement de la présence d'éléments très détaillés mais également de leur très grand nombre. Plusieurs techniques ont été développées pour définir une représentation plus dense des objets comme par exemple les textures volumiques qui sont utilisées pour définir de grandes étendues de végétaux tels que des forêts. Le problème de ces approches réside surtout dans le passage de la vue globale à la vue détaillée. Lorsque l'on s'approche d'un objet contenu dans l'ensemble, la représentation volumique n'est plus suffisante pour retranscrire tous ces détails. Lors du passage d'une représentation à l'autre, un artefact visuel est souvent visible.

En ce qui concerne la production d'une animation, le problème est bien différent. Il s'agit principalement d'être capable de rendre chaque plan de la séquence de préférence en un minimum de temps. Beaucoup de techniques de rendu offline ont été développées pour résoudre ce problème de mémoire.

Contributions

Dans cette thèse, nous nous intéressons à l'ajout de détails dans des scènes naturelles. Comme nous l'avons vu précédemment la plupart des méthodes proposées repose sur un plaquage de textures. Or, cela n'est pas suffisant pour rendre compte de la diversité des formes et des tons que l'on peut observer dans la nature. De même, un plaquage de textures ne permet pas de rendre compte de la complexité des interactions lumineuses qui peuvent intervenir. C'est le cas notamment, de l'auto-ombrage, ou encore du transport de la lumière par les différentes couches de matériaux dont sont constitués les éléments d'une scène naturelle. De plus, cette dégradation est le plus souvent le résultat d'une simulation physique. Cette approche présente pour principal inconvénient de nécessiter de longs temps de calcul et peut être délicate à contrôler lorsque l'on souhaite un résultat précis.

Notre objectif est d'augmenter le réalisme des scènes naturelles. De plus, nous souhaitons nous approcher le plus possible du temps interactif afin d'offrir au designer un retour visuel rapide du résultat du processus d'amplification. Enfin, nous nous plaçons dans le cas général, c'est à dire que les détails doivent être réalistes, qu'il s'agisse d'une vue globale de la scène ou d'une vue rapprochée sur un élément. Pour cela, nous proposons de disperser des éléments géométriques dans la scène. Cette approche pose principalement quatre problèmes qui sont :

- i La modélisation des détails : Pour obtenir une scène réaliste même lors d'une vue rapprochée, il est nécessaire de modéliser de manière très réaliste les détails.
- ii La génération de la variété : Modéliser de manière très précise un détail n'est pas suffisant. Pour que la scène soit réaliste, il faut pouvoir y disperser non pas un détail mais une très grande variété. C'est cette diversité qui apporte la richesse de tons que l'on retrouve dans la nature. Ainsi, même si dans l'image finale, chaque détail ne représente pas plus d'un pixel, la variété permettra d'augmenter considérablement le réalisme de la scène.

- iii Leur répartition dans la scène : La répartition des détails dans la scène est un élément important pour son réalisme final. Il est nécessaire de les placer de manière réaliste de sorte qu'ils ne constituent pas un élément visuellement dérangeant. De plus, dans le cadre d'un travail artistique, il faut fournir au designer des outils de haut et de bas niveaux qui sont des instruments complémentaires. Tandis que les premiers permettent de contrôler la répartition des éléments de manière globale, les seconds sont nécessaires pour la traiter localement.
- iv La gestion de la mémoire : Pour aboutir à une scène réaliste, il est nécessaire d'ajouter un très grand nombre d'éléments de détail. Or, ceci a un coût en mémoire non négligeable dès lors que l'on veut pouvoir naviguer dans la scène.

Notre approche repose sur la définition de trois éléments 1.1 : un environnement, un modèle d'évolution et un modèle de dispersion.

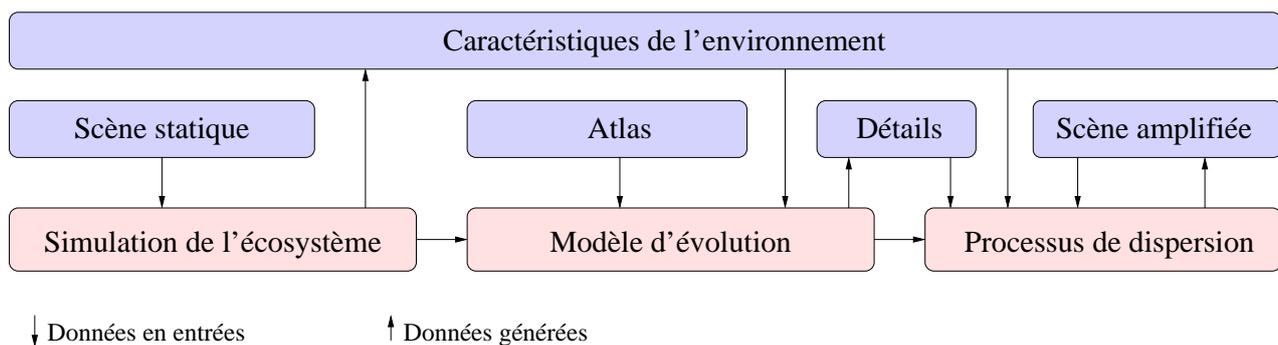


FIG. 1.1 – Principe de notre approche.

Nous utilisons les propriétés de l'environnement pour générer des détails de forme réaliste et les placer de manière plausible dans la scène. Afin de simplifier l'évaluation de ces propriétés et en considérant que les détails sont de petite taille, nous faisons l'hypothèse que leur ajout ne modifie pas l'écosystème, c'est à dire qu'ils n'ont pas d'effet rétroactif. Cela nous permet d'évaluer les données de l'environnement lors d'une phase de pré-calcul ce qui est indispensable afin de réaliser par la suite l'ajout de détails en temps interactif.

Pour modéliser les détails et générer la variété, nous utilisons des fragments organisés dans un atlas en fonction de leurs propriétés. Ces fragments sont ensuite combinés ensemble par un modèle d'évolution. Le détail ainsi obtenu est généré en tenant compte des données de l'environnement et de la surface.

La localisation des détails dans la scène est réalisée par le modèle de dispersion adapté au type de détails à disséminer. Ce processus utilise les données de l'environnement et les propriétés de la surface pour définir les zones sur lesquelles vont se développer les détails.

Organisation du document

Nous commençons ce document par une présentation des travaux réalisés dans le domaine de la modélisation de scènes naturelles et la simulation de phénomènes de dégradation.

Les trois chapitres suivant décrivent comment nous avons utilisé notre approche pour simuler trois phénomènes naturels qui sont : la colonisation du lichen, la formation de fissures et enfin, la simulation de l'évolution et de la chute des feuilles en automne.

Nous présentons notre système de simulation de la propagation de lichen dans le chapitre 3. Le dépôt des lichens est réalisé, soit à partir de l'information d'accessibilité de la surface, soit par le biais d'informations peintes directement dans la scène par l'utilisateur. Les motifs de lichens sont générés par un algorithme de diffusion limité par agrégation que nous avons modifié : L'Open-DLA. Cet algorithme prend en compte les caractéristiques de l'environnement et la compétition de territoire de plusieurs entités lors de la phase de propagation. Enfin, la géométrie finale des lichens est obtenue en utilisant un atlas de modèles texturés qui représentent des fragments de thalles qui sont recombinaés à partir des informations de propagation.

Nous traitons de la simulation de l'évolution et de la chute des feuilles dans le chapitre 4. Les feuilles sont modélisées à partir de feuilles scannées et sont stockées dans un atlas. Nous avons développé un modèle capable de représenter leur évolution au cours du temps et en fonction des caractéristiques de l'environnement. Ce modèle se nomme : l'Open Markov Chain. Pour simuler le dépôt des feuilles dans la scène, nous utilisons un ensemble pré-calculé de trajectoires de chutes. Une fois que les feuilles sont empilées, nous utilisons une simulation physique pour stabiliser les tas.

Dans le chapitre 5, nous détaillons notre système d'ajout de fissures. Notre approche consiste à éditer des motifs de fissures à partir de photographies et à les stocker dans un atlas. Ces motifs en deux dimensions, sont ensuite projetés sur un objet afin de générer une fissure en trois dimensions. L'objet fissuré est alors obtenu par différence booléenne entre l'objet initial et le volume de la fissure. Nous avons défini le modèle d'évolution associé aux fissures en paramétrant le motif par le temps. Dans le cas des fissures, il est difficile de définir un algorithme de dissémination. Nous avons développé des outils qui permettent à un designer de les placer précisément directement sur les objets.

Enfin, dans le chapitre 6, nous faisons le point sur les apports de notre approche et ses limites.

Chapitre 2

Etat de l'art

Sommaire

2.1	Processus de génération de scènes naturelles	9
2.1.1	Génération de terrains	9
2.1.2	Modélisation des végétaux	11
2.1.3	Dissémination de végétaux	13
2.2	Modélisation de détails sur les végétaux	16
2.2.1	Modélisation des poils et des épines	16
2.2.2	Création de l'écorce	17
2.2.3	Modélisation des feuilles et des pétales	19
2.3	Vieillessement	21
2.3.1	Techniques de vieillissement de surfaces	21
2.3.2	Fissures et Fractures	24
2.4	Synthèse	30

Ce chapitre est décomposé en trois parties. Tout d'abord, dans la partie 2.1, nous présentons le processus de modélisation de scènes naturelles complexes et présentons un état de l'art des techniques de modélisation, de terrains, de végétaux et les méthodes de simulation d'écosystèmes. La section 2.2 présente les techniques d'ajout de détails sur les végétaux. Enfin, la section 2.3 traite des techniques de vieillissement.

2.1 Processus de génération de scènes naturelles

En 1985, Reeves et Blau [80] proposent un processus de modélisation adapté au problème des scènes naturelles. Ce processus, repris par Chiba [24] se décompose en quatre étapes :

- La première étape consiste à modéliser le terrain.
- Les végétaux sont ensuite placés dans la scène. Deux méthodes sont proposées, l'une procédurale, l'autre interactive.
- La troisième étape consiste à modéliser les plantes individuelles. Pour Reeves et Blau, il s'agit des arbres et de l'herbe par exemple.
- La dernière étape est l'étape de rendu.

Ce processus met en évidence les différentes problématiques de modélisation d'un paysage naturel. Tout d'abord, le premier problème est celui de la modélisation du terrain. Ensuite, comment placer les plantes de manière automatique ou semi-automatique. Etant donné la complexité d'un écosystème, il n'est pas envisageable de le faire à la main à moins de disposer d'outils adaptés : comment modéliser ces plantes de manière réaliste ? Dans la nature, les plantes arborent des structures complexes et font preuve d'une très grande diversité : comment reproduire cette diversité et surtout, comment la gérer ? La gestion de la mémoire est donc un aspect non négligeable dans la modélisation d'un paysage virtuel : il faut avoir des structures de données adaptées à ce problème. Enfin, comment faire le rendu d'une scène forcément très complexe dans un temps raisonnable contenu de la quantité d'informations que représente un paysage naturel synthétique ?

2.1.1 Génération de terrains

La génération de terrain est une étape importante dans la modélisation d'un paysage. La forme du relief donne des informations sur la nature et l'histoire du sol. Certains travaux dans ce domaine portent sur la modélisation d'un terrain obtenu par le biais de mesures réelles. Dans cette partie, nous ne nous intéressons qu'aux méthodes de génération de reliefs artificiels (Figure 2.1). Les approches qui sont généralement utilisées peuvent être classées en deux catégories : les approches procédurales et les méthodes utilisant une simulation physique. Nous allons décrire chacune de ces approches en présentant leurs avantages et leurs inconvénients.



FIG. 2.1 – A gauche : Exemple de terrain généré par MojoWorld [28]. Au milieu : "Medicine Lake", paysage produit par Musgrave par une méthode multi-fractale [28]. A droite : Exemple de terrain avec des rivières créés par méthode fractale [76].

2.1.1.1 Méthodes procédurales

Dans les années 1968, B. Mandelbrot met en évidence la similitude entre les crêtes des montagnes et la courbe produite par un mouvement Brownien fractionnaire appelé fBm. Il s'agit d'un ensemble de petits mouvements fractionnaires indépendants et isotropes. Il publie en 1982 le livre *The Fractal Geometry of Nature* [54] dans lequel il montre l'intérêt d'utiliser les fractales pour représenter les formes naturelles. La synthèse de fréquence spectrale est communément utilisée dans la littérature pour calculer une approximation du fBm [55]. Cette méthode repose sur la transformée inverse de Fourier pour convertir la fréquence en altitude. Le principal défaut de cette méthode est qu'elle présente de fortes symétries sur les plans horizontaux et verticaux. De plus, elle génère des terrains présentant des motifs périodiques. Cette approche permet un bon contrôle global de la forme du terrain mais n'offre pas un bon contrôle local des détails. Le calcul de la fBm est compliqué et surtout très coûteux en temps de calcul. Afin de résoudre ce problème, plusieurs méthodes stochastiques ont été proposées. Elles reposent sur une subdivision récursive de la surface combinée avec l'ajout d'un facteur aléatoire. On peut citer parmi les plus connues : le *diamond-square* de Miller [60], le *square scheme* de Fournier [36] et la subdivision hexagonale de Musgrave [74]. La plupart de ces méthodes ne sont qu'une approximation de la fBm et ne vérifient pas les mêmes caractéristiques d'isotropie et d'autosimilarité ce qui produit des ruptures de la continuité du relief et des transitions brutales. Afin de résoudre ces problèmes, des méthodes tenant compte du voisinage lors de la subdivision ont été introduites [52, 60]. Les méthodes stochastiques sont faciles à mettre en place mais elles ne permettent pas un bon contrôle local de la surface. Voss et Mandelbrot [89] ont proposé des méthodes de subdivision par ajouts de matière successifs aléatoires. Dans ces méthodes, on commence avec un terrain plat et on augmente de manière répétitive l'altitude sur une portion du terrain. L'algorithme le plus courant utilise une ligne de démarcation générée aléatoirement sur le terrain. L'altitude est alors augmentée à droite ou à gauche de cette ligne. J. Olsen a proposé une variation de cette méthode dans laquelle la région à modifier est définie par de grands disques. L'avantage de ces méthodes est que le niveau de détails dépend directement du nombre d'itérations, ce qui permet un bon contrôle de la finesse du terrain. Toutefois ces

méthodes sont coûteuses en temps de calcul. Dans la nature, les terrains ne sont pas identiques sur toute leur surface. Les reliefs âgés sont plus réguliers que les crêtes des montagnes jeunes. Les méthodes fractales précédentes produisent des terrains présentant des similitudes à différentes échelles et positions. Or, l'invariance d'échelle et l'autosimilarité sont deux propriétés importantes des fractales. Musgrave [65] a proposé une méthode multi-fractales permettant de reproduire ce phénomène. Une multi-fractale est définie comme la somme ou le produit de fonctions mono-fractales comme les fBm. Cette approche permet de générer des terrains avec des zones lisses et des zones plus rugueuses. Toutefois, elle est plus coûteuse en temps de calcul que les méthodes mono-fractales.

2.1.1.2 Méthodes de simulation de terrains par érosion

Des méthodes s'appuyant sur les lois de la physique ont été proposées pour augmenter le réalisme des terrains générés par les méthodes précédentes. La plupart des algorithmes étudiés utilisent les lois de l'hydrodynamique et des modèles de sédimentation. Musgrave [65] a proposé deux algorithmes de simulation d'érosion de terrains : le premier simule le mouvement de sédiments provoqué par des chocs thermiques ; le deuxième simule la dissolution et le dépôt des sédiments par l'eau. Chiba [24] et Nagashima [66] ont proposé un algorithme d'érosion reposant sur l'action de dissolution des roches par l'eau. Ils utilisent un système de particules pour représenter la course de l'eau. Benes [12] a proposé une méthode hybride reposant sur le même principe mais ne vérifiant pas les lois de la physique. Cette méthode présente l'avantage d'être interactive. D'autres travaux ont été proposés afin d'ajouter des rivières aux terrains [76, 9].

2.1.2 Modélisation des végétaux



FIG. 2.2 – A gauche : *Xerophyllum tenax* modélisé par l'approche décrite dans [79]. Au milieu : Bonsaï modélisé à partir d'un graphe de décomposition [19]. A droite : Modèle obtenu en contraignant une plante dans un volume en forme de dinosaure [77].

La modélisation de végétaux réalistes pose deux problèmes fondamentaux. Le premier problème est celui de la complexité des formes et des textures à modéliser. Le second problème est celui posé par la variété des détails qui est importante et diffère selon les espèces. Dans cette partie, nous nous intéressons aux modèles qui ont été développés pour représenter la structure des plantes. Ces modèles peuvent être classés selon l'approche qu'ils utilisent. Il existe deux types d'approches : une approche descendante et une approche montante.

2.1.2.1 Approche descendante

Cette approche consiste à définir la structure de la plante ou de l'arbre à partir d'une forme simple. Cette forme définie en principe la silhouette globale de la plante. Le branchage est ensuite construit de sorte qu'il soit inclus dans celle-ci. La méthode la plus employée dans cette approche est la modélisation procédurale. Le programme *TreesDesigner* développé par Pawel Olas est un bon exemple de cette approche. A partir de quelques paramètres, il construit une structure dans laquelle les branches se terminent sur le bord de la silhouette définie en entrée. Cette méthode a été décrite par Reeves et Blau [80] puis étendue par Weber et Penn [91], Linterman et Deussen [30, 16] et Prusinkiewicz [79]. Le gros problème de ces méthodes est que plus l'on veut obtenir une plante réaliste et plus on doit déterminer de paramètres pour pouvoir définir les variations entre chaque niveau et à l'intérieur d'un même niveau. Une approche permettant de résoudre cet inconvénient a été proposée par Boudon dans [19]. Elle consiste à définir un graphe de décomposition. Ce graphe décrit les caractéristiques de chaque niveau et comment passer de l'un à l'autre. Les paramètres nécessaires à la génération de la plante sont ensuite obtenus à partir du graphe.

L'avantage de cette approche est que la modélisation est très intuitive. La plante va correspondre à la silhouette définie en entrée. Un des inconvénients est qu'elle offre peu de contrôle au niveau de la répartition des sous-branches.

2.1.2.2 Approche montante

Ce type d'approche correspond mieux à ce qui se passe dans la nature. La structure de la plante est définie de manière progressive en partant du tronc. Les méthodes les plus utilisées sont les méthodes à base de subdivisions et les méthodes reposant sur la réécriture de règles.

En 1988, Reffye et *al.* [26] proposent une méthode simulant le cycle de développement d'une plante. Celle-ci est considérée comme un ensemble de bourgeons pouvant devenir une fleur, un embranchement de plusieurs tiges, mourir ou bien rester dans le même état. Le passage d'un état à l'autre est guidé par une fonction de probabilité qui dépend, entre autres, de l'âge et de la taille de la tige concernée.

En 1968, Lindenmayer introduit un modèle permettant de décrire le développement d'organismes multicellulaires : les L-Systemes. Ces modèles étaient initialement prévus pour décrire le développement de cellules individuelles. Frijters et Lindenmayer les ont ensuite étendus

à la description du développement d'organismes plus complexes comme les plantes. Un L-Systemes est représenté par un ensemble de règles et de symboles définissant une grammaire. Les règles du L-Systeme sont appliquées itérativement à partir de l'état initial. Dans le modèle proposé par Frijter et Lindenmayer, les règles de production font référence à un seul élément et ne font intervenir aucune information supplémentaire. Cette approche permet de générer des plantes ayant un branchage régulier. Le problème est que dans la nature, les facteurs extérieurs jouent un rôle important dans la forme finale de cette structure.

Afin de résoudre ce problème, deux extensions ont été proposées. Les L-Systemes paramétriques et les L-Systeme dépendants du contexte ont été introduits dans les années 1990. Ces systèmes sont détaillés dans *The algorithmic beauty of plants* [78]. Le principe est d'ajouter des paramètres de contrôle aux règles de production et d'inclure une dépendance des prédécesseurs. De nombreux travaux exploitent ces modèles pour simuler la compétition pour des ressources telles que l'eau et la lumière [13, 11]. La structure de la plante est alors beaucoup plus réaliste. Cette approche permet d'avoir un bon contrôle au niveau de la répartition et du développement des branches. Toutefois, le problème de ces méthodes est qu'elles offrent peu de contrôle de la forme finale.

Certains travaux, comme [77], utilisent cette représentation pour contraindre la plante à rester à l'intérieur d'un espace virtuel, lui conférant ainsi une forme particulière. De même, certaines approches hybrides, tel que le système de modélisation XFrog, tirent avantage des deux approches : procédurale et à base de règles. Dans ce système, la plante est définie par un graphe. Les nœuds et les feuilles contiennent des procédures de modélisation.

2.1.3 Dissémination de végétaux

La distribution de plantes dans une scène naturelle est généralement réalisée selon trois approches (Figure 2.3). La première approche consiste à simuler certains phénomènes à l'échelle de la plante comme la compétition pour les ressources : la distribution est alors obtenue comme un phénomène émergent. La deuxième approche consiste à utiliser une carte de densité à partir de laquelle la distribution va être générée. Enfin, la dernière approche consiste à utiliser les méthodes de pavage pour générer l'écosystème.

2.1.3.1 Simulation d'écosystème

Les premiers travaux sur la simulation d'écosystèmes ont été présentés par Aikman et Watkinson [6]. Ces travaux sont repris dans le domaine de l'informatique graphique par Deussen [29]. Le principe est le suivant : une plante est représentée par un disque définissant sa zone d'influence. La simulation est initialisée par une distribution aléatoire de disques de tailles différentes. Si des disques s'intersectent, la plante qui présente la plus petite capacité de compétition meurt. Cette capacité est définie par un ensemble de paramètres décrits dans [29]. De



FIG. 2.3 – A gauche : Ecosystème complexe produit par la méthode présentée par Deussen dans [29]. Au milieu : Exemple de distribution de plantes obtenue par la méthode de Lane [50]. A droite : Distribution de tournesols générée par le modèle de pavage décrit dans [25].

plus, la plante meurt lorsqu'elle atteint sa taille adulte. Cet algorithme permet à Deussen de reproduire le phénomène d'éclaircissage présent dans la nature. Le problème de cette approche est qu'elle produit une distribution uniforme de plantes. Une des raisons est que seule une plante est considérée durant le processus de croissance et non un groupe.

Lane [50] a proposé un modèle permettant de tenir compte du développement d'un groupe de plantes. Dans la nature, on peut observer que les plantes de la même famille ont tendance à se développer en touffe. Ce phénomène a un impact important sur l'aspect de la distribution. Le *multiset L-system* défini par Lane est une généralisation du L-System qui permet de définir des règles de croissance pour un groupe de plantes. Dans [50], Lane utilise son système pour décrire trois phénomènes importants dans le développement d'un groupe de plantes. Le premier est l'éclaircissage. Ce phénomène correspond à l'étouffement et éventuellement à la mort des jeunes plantes dominées par les plus grosses lors de la compétition aux ressources. Le deuxième phénomène important est la succession : lorsqu'une plante meurt, elle est remplacée par une plante plus petite. Sans prise en compte de la succession, cette plante disparaîtrait suite au phénomène précédent. Enfin, le dernier phénomène simulé est la propagation : une plante donne naissance à un ensemble d'autres dans son voisinage.

Le principal inconvénient de ces travaux est qu'ils ne simulent qu'un aspect de la compétition pour les ressources. Si l'on considère deux plantes bénéficiant des mêmes ressources soit les plantes se partagent équitablement les ressources, soit la plante la plus grosse les monopolise. Alsweis et Deussen [7] proposent un modèle qui prend en compte ces deux aspects de la compétition. Les simulations qu'ils produisent montrent l'importance de cette dualité qui permet d'obtenir des écosystèmes très réalistes.

Ces techniques ne permettent pas de contrôler la distribution qui est le résultat de la simulation de phénomènes à l'échelle individuelle. Benes [15] propose un modèle s'appuyant sur la notion d'agents agissant comme des jardiniers par exemple pour désherber une zone de culture. L'écosystème est simulé par une version étendue de l'algorithme de Lane décrit dans [14]. En affectant des tâches aux agents, il est possible de contrôler la distribution de plantes.

2.1.3.2 Distribution de plantes à partir d'informations de densité

Les méthodes reposant sur la simulation de comportement individuel offrent peu de contrôle. Le résultat de la simulation n'est pas prédictible. Une autre approche couramment utilisée, en particulier pour la création de forêts, consiste à utiliser une carte de densité. Dans ce cas précis, on cherche à obtenir une densité de distribution uniforme avec un écart entre les arbres à peu près constant. La difficulté de cette approche réside dans l'instanciation des végétaux à partir des informations de densité. Chiba [24] utilise une méthode stochastique qui repose sur l'échantillonnage d'une surface par des disques. Cette méthode génère une distribution régulière ce qui est rarement le cas pour un grand nombre d'espèces.

Dans [29] Deussen propose d'utiliser des cartes de densités créées à partir de données réelles ou bien peintes à la main. Il utilise un algorithme de *demi-ton* pour le processus d'instanciation. Le réalisme de la distribution est donc fortement dépendant des données récoltées. De plus, cette méthode ne permet pas de contrôler l'éclaircissage de manière réaliste. Dans [50], Lane définit un algorithme d'instanciation permettant de tenir compte de l'éclaircissage. Cet algorithme repose sur une probabilité de présence d'une plante. Chaque fois qu'une plante est ajoutée, elle modifie dans son voisinage la probabilité qu'une autre plante soit présente, ceci en respectant la densité voulue. Cette approche permet de définir le seuil d'éclaircissage entre chaque plante et de contrôler leurs interactions.

2.1.3.3 Pavage

Les méthodes de pavage [25, 38] consistent à considérer non plus les plantes de manière individuelle mais comme un groupe. L'objectif est de parvenir à distribuer ces groupes de plantes sans faire apparaître de discontinuité ou de symétries trop visibles. On peut citer entre autres Neyret [67] et Cohen [25] qui ont proposé des méthodes permettant de générer des distributions non-périodiques.

Ces approches sont bien adaptées à la création de grande étendues constituées par des végétaux homogènes. C'est par exemple le cas des champs cultivés mais aussi des étendues de gazon. Le principal avantage de cette méthode c'est qu'elle permet d'utiliser peu de mémoire car ce sont les mêmes modèles qui sont réutilisés en différents endroits de la scène. De plus, cela permet de décrire la scène avec très peu de paramètres. Un inconvénient majeur de cette approche vient du fait que de par sa nature, elle ne peut s'appliquer qu'à des étendues homogènes.

2.1.3.4 Conclusion

Dans les travaux précédents, on peut constater que deux approches sont principalement utilisées pour ajouter de nouveaux éléments dans la scène. La première méthode (approche montante) consiste à partir de la donnée de l'élément pour aboutir à la notion de population. La deuxième méthode (approche descendante) consiste à partir de la donnée d'une population

pour aboutir à la notion d'individus.

Le principale avantage de l'approche montante provient du fait qu'il s'agit d'une approche naturelle du problème. Dans le cas de détails biologiques, les paramètres couramment utilisés sont les caractéristiques biologiques de l'élément. Toutefois, dans certains cas, ceux-ci peuvent ne pas être connus. De plus, cette approche ne permet pas de contrôler de manière globale le résultat voulu, on manipule uniquement les briques de bases. La distribution finale est le résultat d'un phénomène émergent. De ce fait, il peut être difficile de reproduire une scène précise.

A contrario, l'approche descendante passe par un contrôle global de la forme du résultat. Ainsi, il est possible de spécifier l'apparence générale de la scène. Les paramètres les plus utilisés sont ceux qui permettent de définir la densité de présence des éléments considérés dans la scène. Cette méthode est particulièrement adaptée à la reproduction d'une scène réelle. Toutefois, dès que l'on souhaite quelque chose de précis, il faut injecter un grand nombre de données à des échelles de plus en plus proches de l'échelle de l'élément. Un des principaux inconvénients de cette approche est qu'elle permet de définir un résultat final contre nature. Par exemple, si l'on considère la forme d'un arbre, son allure doit être corrélée avec les sources de lumières disponibles.

Dans le cas de la problématique de l'ajout de détails, il faut pouvoir combiner ces deux approches. Pour cela, nous proposons d'utiliser conjointement deux algorithmes : un algorithme de dispersion et un algorithme d'évolution. L'algorithme de dispersion pourra utiliser indifféremment l'une ou l'autre des approches. L'algorithme d'évolution quant à lui garantira que le résultat ne sera pas contre nature en ne développant pas les éléments qui ne pourraient pas l'être selon les conditions dans lesquels ils se trouvent.

2.2 Modélisation de détails sur les végétaux

Dans cette partie, nous allons nous intéresser aux méthodes qui visent à augmenter le réalisme des plantes en ajoutant des détails sur leur structure comme des poils ou des épines (Figure 2.4). Nous verrons également comment sont modélisés les différents organes constituant les plantes tel que les feuilles et les pétales.

2.2.1 Modélisation des poils et des épines

Dans la nature, les plantes ne sont pas parfaitement lisses. Certaines, comme les fougères mais aussi les crocus, présentent des poils. D'autres espèces telles que les roses ou les cactus, présentent des épines. Plusieurs travaux ont été menés pour représenter ces éléments afin d'augmenter le réalisme des plantes de synthèses.

Fowler *et al.* [37] ont proposé une méthode permettant de décrire la répartition des organes (pétales, étamines) sur la plante en représentant les caractéristiques phyllotaxiques des plantes. Le support des organes est représenté par une surface de révolution. Les organes sont ensuite



FIG. 2.4 – Fougère poilue modélisée par la méthode proposée par Fuhrer dans [53]. Exemple de feuilles présentant des nervures produites par l’algorithme de Runions [81]. Ecorce d’arbre réaliste générée par Wang [90].

placés tangentiellement les uns par rapport aux autres en parcourant la surface de révolution du haut vers le bas. Un algorithme de collision permet de créer une distribution compacte. Fowler et *al.* utilisent cette méthode pour modéliser une grande variété de cactus. La répartition des épines est réalisée à partir du modèle de collision. Cette approche permet de générer des plantes dont les organes sont distribués de manière réaliste. Le seul problème de celle-ci est qu’elle utilise comme support une surface de révolution. Or, dans la nature, les plantes présentent toujours de légères asymétries. De plus, la distribution des épines sur un cactus n’est pas forcément parfaite. Celle-ci va dépendre de son environnement.

Fuhrer et *al.* [53] proposent une méthode pour modéliser les plantes à poils. Cette méthode repose sur les travaux déjà réalisés dans le domaine de la modélisation de fourrures. Les poils sont représentés par des cylindres généralisés qui sont distribués sur la surface de la plante. Fuhrer utilise une fonction de diffusion pour propager un ensemble de points d’attachement de poils sur la plante. La densité, la taille et l’orientation des poils sont directement contrôlées par le L-Systeme. La distribution est contrôlée par une loi de Poisson. Cette approche est purement empirique, cependant les résultats obtenus sont convaincants. Un des inconvénients de cette méthode est que l’utilisation d’un cylindre généralisé par poils génère des modèles très complexes.

2.2.2 Création de l’écorce

Les méthodes de modélisation de l’écorce d’un arbre peuvent être classées en deux grandes familles : l’habillage de texture et la simulation.

2.2.2.1 Approche par création de textures

La première famille consiste à utiliser des textures pour recréer l’apparence de l’écorce d’un arbre. Différents types de textures sont utilisés. Bloomenthal [17] utilise une carte de plaquage de relief pour recréer l’illusion de la rugosité d’une vraie écorce. Il obtient cette texture à partir

d'un moulage de plâtre d'une vraie écorce qu'il a ensuite passée au scanner. Maritaud et *al.* [56] utilisent une carte de déplacement obtenue à partir de la photographie d'une écorce. La texture originale est utilisée sous la forme d'une texture 2D pour décorer l'écorce. L'avantage de cette méthode par rapport à la précédente est qu'elle permet d'obtenir un vrai déplacement de la surface. Le résultat est ainsi plus réaliste, notamment le long de la silhouette des branches. Wang et *al.* [90] utilisent également une carte de hauteur pour représenter le relief de l'écorce. L'originalité de leur méthode vient de la manière dont ils l'obtiennent. A partir d'une photographie d'écorce, ils construisent un ensemble de masques représentant chacun une caractéristique de l'écorce comme par exemple les fractures verticales et horizontales. Le designer peut alors créer la carte de hauteur correspondant au type d'écorce qu'il souhaite en mélangeant les différents masques précédemment obtenus.

Hart et Baker [43] utilisent une texture procédurale produite à partir d'une fonction de bruit. Dans ces travaux, les branches sont définies par des cônes implicites. Cette approche leur permet de mélanger la texture de la branche avec celle de la branche. Les résultats qu'ils obtiennent sont intéressants mais ne sont pas très esthétiques. Le choix de la fonction de bruit en est l'une des raisons.

Le réalisme apporté par ces méthodes dépend principalement de deux facteurs. Tout d'abord, le choix de la paramétrisation. En effet, la variation du diamètre du tronc et les zones de naissance des branches produisent des discontinuités avec une paramétrisation classique, du type cylindrique par exemple. D'autre part, le réalisme dépend également de la taille de la texture. Plus la texture sera grande et moins il y aura de risque de voir apparaître des discontinuités dues à un pavage de celles-ci. Certaines méthodes de synthèse de textures existent afin de résoudre ce problème mais elles ne fonctionnent pas avec tous les types d'écorces.

2.2.2.2 Approches par simulation

La simulation de la formation de l'écorce, contrairement aux méthodes précédentes, évite de passer par une étape de paramétrisation. Dans [34, 35], Federl et Prusinkiewicz proposent un modèle reposant sur les éléments finis. La fissure est générée en retirant du modèle les éléments dont le stress dépasse une valeur seuil. Comme cette méthode nécessite que les éléments soient petits relativement à la taille de la fissure, Federl et Prusinkiewicz proposent un algorithme de subdivision adaptatif. Les éléments traversant le plan de fracture ne sont supprimés que si leur taille est suffisamment petite. Sinon, l'élément est subdivisé, puis le processus recommence. Cette méthode permet d'obtenir une écorce assez réaliste. Toutefois, son principal inconvénient, est qu'elle génère un très grand nombre d'éléments.

Lefebvre et Neyret [51] simulent la croissance de l'écorce avec un système hybride reposant sur les lois de la physique. Leur méthode consiste à décrire l'écorce comme un ensemble de bandelettes circulaires le long de la branche. L'apparition des fractures est obtenue en simulant les tensions générées par la croissance des bandelettes. La propagation des fractures se fait dans la direction orthogonale à celle-ci. Cette méthode permet de générer des écorces réalistes

sans discontinuité au niveau du branchage. De plus, elle fonctionne en temps interactif ce qui permet à un designer de retoucher l'aspect de l'écorce comme il le souhaite.

2.2.3 Modélisation des feuilles et des pétales

Les feuilles sont présentes partout dans la nature. Leur modélisation est une tâche importante en informatique graphique dans la quête du réalisme. De nombreux travaux ont été menés depuis les années 1980 pour modéliser cet élément. On peut les classer en deux familles : la première famille de méthode s'attache à construire la forme de la feuille ; à contrario, la deuxième famille cherche à construire les nervures de la feuille pour en déduire sa forme finale.

Dans la première approche, deux méthodes sont principalement utilisées. Une première technique consiste à construire la feuille à partir d'une photographie. Dans [17], Bloomenthal utilise une photographie pour définir la texture de la feuille. Celle-ci est projetée sur un polygone. Le détourage de la feuille est donné par un masque de visibilité. Avec cette méthode la feuille est plate. Bloomenthal accentue les veines afin de leur donner plus de relief. Les résultats obtenus sont toutefois acceptables du fait de la qualité de la texture. Une autre technique consiste à générer la feuille dans son intégralité. Prusinkiewicz et *al.* [75] proposent d'utiliser les L-Systems pour définir le contour de la feuille. La capacité des L-Système à reproduire les autosimilarités rend cette méthode bien adaptée à la modélisation du contour de la feuille. Toutefois, cette approche a tendance à générer des formes qui présentent des structures répétitives. De plus, cette méthode ne résout pas le problème du manque de relief de la feuille. Lars Mündermann et *al.* [64] proposent un système pour modéliser les feuilles tordues. Le système prend en entrée le contour d'une feuille. À partir de celui-ci, ils construisent interactivement le squelette de la feuille. Ce squelette est en fait son axe médian. Les déformations sont appliquées sur le squelette, puis la surface de la feuille est reconstruite par un procédé de balayage de contours. Cette méthode permet de déformer de façon très naturelle les feuilles en manipulant l'axe médian.

L'autre approche consiste à modéliser les nervures de la feuille. En effet, on peut considérer que la forme de la feuille est très liée à la répartition des nervures. Plusieurs méthodes ont été proposées afin de générer les nervures primaires et secondaires constituant le squelette de la feuille. Ces méthodes reposent sur la construction d'un contour implicite à partir du squelette formé par les nervures principales et secondaires. Cette approche est décrite pour la première fois dans [88]. Viennot et *al.* modélisent les nervures primaires et secondaires à partir de leur nombre et de l'angle entre elles. Hammel et *al.* [42] proposent une méthode similaire : ils utilisent un L-Système pour décrire la structure de la feuille. La forme finale est obtenue en associant à chaque partie du squelette une fonction de potentiel décrivant sa forme. L'avantage de cette approche est que les méthodes produisent une description procédurale de la feuille. Ceci permet de représenter une feuille par un minimum de paramètres. Par contre, pour obtenir une feuille colorée, il est nécessaire de générer sa texture. Ces dernières, présentent généralement moins de richesses visuelles qu'une photographie mais restent réalistes dans des vues

éloignées.

Afin d'augmenter le réalisme de l'aspect des feuilles, des travaux ont été menés dans le domaine de la génération de veinures. Contrairement aux travaux précédemment cités, l'objectif n'est pas de produire un squelette pour générer la silhouette de la feuille. Le but est de la décorer. De plus, le squelette correspond à l'axe médian de l'objet. Or, cet axe médian ne coïncide pas forcément avec les veinures principales des feuilles.

Runion et *al.* [81] ont développé un modèle dans lequel ils simulent la formation des veinures durant la croissance de la feuille. Leur modèle repose sur la simulation de la diffusion et de l'absorption des hormones responsables de la formation des veinures : les auxines. Les résultats obtenus sont très réalistes et permettent d'augmenter de manière significative le réalisme des feuilles synthétisées.

2.2.3.1 Conclusion

En étudiant les travaux portant sur la modélisation des détails des végétaux, on constate qu'en général, les détails microscopiques tels que les veinures sont modélisés par le biais de textures, voire dans certains cas, une texture associée à une carte de BRDF. Par contre, dans le cas de détails macroscopiques, tels que l'écorce des arbres, les méthodes proposées génèrent souvent la géométrie. Celle-ci est également parfois associée à une texture.

Dans cette thèse, nous nous intéressons à l'ajout de détails dans son ensemble, et ce dans l'optique d'une vue rapprochée. Notre approche repose sur le fait qu'un détail est un élément de décoration. A la lumière des travaux précédents, nous allons nous placer dans une problématique d'habillage de géométries, ceci de manière similaire à l'habillage de textures. Les éléments de la scène vont être considérés comme des supports sur lesquels on va produire de la géométrie. Cette approche s'apparente d'avantage à la synthèse de texture qu'à la simulation physique qui consiste en général à raffiner localement le maillage de la surface. Ce choix vient principalement du constat qu'un plaquage de textures seul est insuffisante pour représenter un élément macroscopique, dès lors qu'il atteint une certaine taille. Ainsi, une texture associée à une carte de normale ne sera plus suffisante dans le cas d'une vue rapprochée, ou encore pour générer un auto-ombrage réaliste. De plus, il est toujours possible de générer une texture associée à une carte de BRDF à partir de la géométrie, même si celle-ci représente le niveau microscopique. Cette méthode offre donc le maximum de représentabilité.

Cette approche pose toutefois le problème de la gestion de la mémoire. En effet, si l'on commence à distribuer une grande quantité d'éléments dans la scène, on va alourdir de manière dramatique sa représentation, et notamment la quantité de mémoire qu'elle utilise. Pour résoudre ce problème, nous proposons d'utiliser la même approche que celle actuellement utilisée pour les textures qui est l'instanciation. Nous allons créer un atlas qui stocke des éléments géométriques simples qui seront combinés (algorithme d'évolution) et instanciés dans la scène pour produire la forme complexe du détail. La diversité nécessaire au réalisme pourra alors être obtenue par le biais de techniques telles que le morphing entre les éléments de l'atlas.

2.3 Vieillessement

Les techniques de vieillissement ont pour objectif d'ajouter des imperfections sur les objets de synthèse généralement trop parfaits. Badler et Becket [8] ont été les premiers à montrer que l'aspect très lisse des objets virtuels trahit leur nature synthétique.

Dans cette partie, nous nous intéressons à deux familles de dégradations. La première famille regroupe les imperfections qui ont lieu à la surface des objets mais n'entraîne pas de profonds changements topologiques. La deuxième grande famille est constituée par les grandes dégradations que sont les fissures et les fractures qui peuvent générer des changements de topologie.

2.3.1 Techniques de vieillissement de surfaces



FIG. 2.5 – A gauche : Simulation de l'accumulation de phénomènes de vieillissement (mousse, pollution,..) sur une bouche d'incendie produit par l'utilisation de γ -ton [23]. Au milieu : Simulation de l'apparition de patines sur un Bouddha en cuivre [32]. A droite : Brouette rouillée générée par l'algorithme de corrosion proposé par Merillou et *al.* [58].

Dans la nature, les objets sont soumis à divers phénomènes naturels qui conduisent généralement à leur dégradation. Parmi ces phénomènes, le transport de matière par l'eau et les dégradations chimiques comme la formation de patine et la corrosion (figure 2.5 et 2.6), ont été largement étudiés dans le domaine de l'informatique graphique.

2.3.1.1 Salissures

En 1996, Dorsey et *al.* [33] ont développé une méthode de vieillissement qui permet de simuler le transport des saletés par la pluie ainsi que leur dépôt. Cette méthode repose sur une simulation physique de l'écoulement de l'eau. Celle-ci produit une carte de textures et n'entraîne aucune modification géométrique de l'objet. La simulation tient compte de la rugosité

des matériaux et de leur taux d'absorption. Cette méthode permet d'augmenter de manière significative l'aspect réaliste des objets.

Aucun phénomène d'érosion n'est simulé dans les travaux précédents. Or, dans la nature l'érosion joue un rôle important dans l'aspect des objets situés à l'extérieur comme les statues et les murs. En 1999, Dorsey et *al.* [31] définissent un modèle permettant de simuler le transport de l'eau, la dissolution et la recristallisation des minéraux en fonction de la porosité de la roche. Le système repose sur la description d'un modèle de pierre qui permet une modification géométrique de l'objet. La méthode de rendu intègre une méthode de diffusion de la lumière dans un matériau opalescent qui augmente nettement le réalisme des résultats. Un des points forts de ce modèle est qu'il permet de simuler des phénomènes chimiques qui ont lieu à l'intérieur de la pierre.

Les travaux de Chen et *al.* [23] généralisent ce phénomène de transports de sources de vieillissement telles que les particules de rouilles, les mousses mais aussi la poussière. Cette technique est désignée par le terme $\gamma - ton$. Le principe utilisé est similaire à celui du tracé de photons. La différence vient du fait que ce n'est plus une quantité d'énergie qui est véhiculée mais des éléments responsables du vieillissement. Les $\gamma - ton$ sont lancés dans la scène suivant des directions probabilistes et ils vont successivement capturer et déposer de la matière sur leur passage. Les résultats obtenus par cette approche sont très réalistes. L'utilisation de cette méthode permet de simuler le vieillissement causé par plusieurs sources de vieillissement comme la rouille, les mousses et la poussière. De plus, contrairement à la méthode proposée par Dorsey dans [33], l'utilisation des $\gamma - ton$ permet de modifier la géométrie de l'objet. Toutefois, Chen et *al.* utilise pour cela une carte de déplacement qui est modifiée dynamiquement par les $\gamma - ton$ à chaque pas de la simulation. Cette représentation ne permet pas entre autre de représenter la formation de trous dus à un transport de matière important.

2.3.1.2 Les patines métalliques

Les méthodes proposées pour modéliser la formation de patine sur des objets sont principalement des méthodes empiriques. La formation de patine est réalisée sur la texture de l'objet. Sa géométrie est inchangée. Dorsey et *al.* [32] propose une méthode procédurale. La formation de patine est produite dans l'espace texture par un ensemble d'opérations de vieillissement contrôlées par l'utilisateur. Le réalisme des résultats obtenus par cette méthode dépend du don artistique du designer. La méthode développée par Wong et *al.* dans [93] repose sur l'estimation de l'accessibilité de la surface et sur des facteurs géométriques. Comme dans la méthode précédente, les patines sont stockées sous la forme d'une texture ce qui ne permet pas de modifier la géométrie de l'objet. Les résultats obtenus ne sont pas très réalistes.

2.3.1.3 Oxydation et corrosion

Les travaux traitant de la formation de rouille peuvent être classés en deux catégories. La première catégorie concerne les méthodes empiriques et la deuxième les simulations phy-

siques.

Dans la première catégorie de travaux, la rouille est représentée par une texture et la propagation est généralement réalisée de manière aléatoire. Merillou et *al.* [58] proposent un modèle reposant sur des mesures réelles. La simulation utilise une carte de corrosion qui stocke les propriétés de la surface et notamment l'apparition de trous. La propagation de la rouille est effectuée par marche aléatoire. La méthode ne modifie pas la géométrie de la surface. Toutefois, les trous dus à la corrosion sont stockés. Le rendu se fait par un lancé de rayon virtuel qui utilise la carte de textures pour rendre la vraie surface. De plus, la corrosion modifie localement la BRDF. Les résultats obtenus avec cette méthode sont très réalistes, notamment grâce à l'utilisation de la BRDF.

La deuxième catégorie de méthodes comprend les méthodes reposant sur une simulation physique. Gobron [40] utilise un automate cellulaire pour simuler la propagation de la rouille sur l'objet. La formation de rouille est réalisée à partir de points de rouille initiaux et d'un modèle de sédimentation. Les résultats obtenus par cette approche ne sont pas très réalistes. Chang et Shih [22] ont proposé une méthode simulant l'apparition de rouille sur des objets placés dans l'eau de mer en modélisant l'effet des courants et les échanges avec l'environnement. Cette méthode repose sur un Open L-Système pour simuler les interactions entre les objets et la mer. Ils utilisent le calcul d'une valeur de tendance pour déterminer l'exposition de la surface à l'eau. Le point fort de cette méthode, est que l'apparition et la propagation de la rouille sont guidées par les caractéristiques de l'environnement (courant, salinité). Les résultats obtenus sont réalistes.

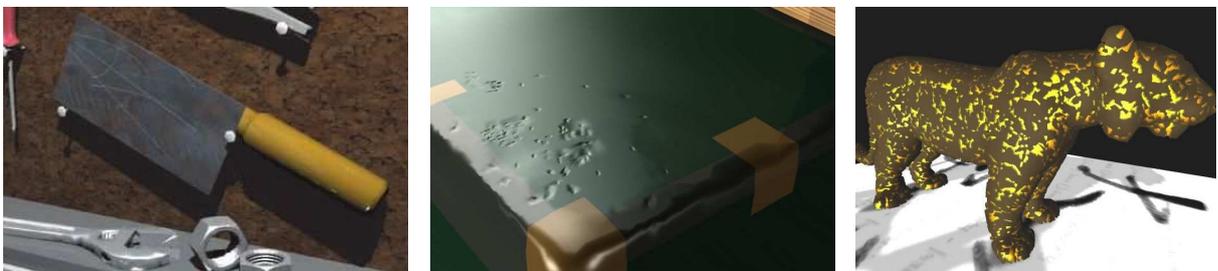


FIG. 2.6 – A gauche : Ajout de rayures à partir de données réelles [57]. Au milieu : Vieillessement par ajout d'impacts [72]. A droite : Simulation de la formation d'écaillures sur une couche de peinture [39].

2.3.1.4 Les rayures

Les rayures sont le résultat de frottements mécaniques. Leur apparition ne peut pas être prédite par le biais d'une simulation physique. La majorité des approches proposées repose sur un modèle empirique qui ne modifie pas la géométrie de la surface. C'est le cas du modèle proposé par Buchanan et Lalonde [21]. Ce modèle repose sur une modification du modèle de Phong dont les paramètres sont guidés par l'observation. Merillou et *al.* [57] proposent un mo-

dèle qui repose sur une carte de BRDF. L'originalité de cette approche provient du fait que cette carte est obtenue en mesurant de vraies rayures. Ce modèle permet d'obtenir de bons résultats. Merillou et *al.* [18] étendent leur modèle en simulant le processus de formation des rayures. Cette étape supplémentaire permet d'obtenir un aspect plus réaliste en affinant le mélange de la BRDF de la surface de l'objet et celle de la rayure le long de celle-ci.

2.3.1.5 Les impacts

Dans la nature, de nombreuses surfaces présentent des traces d'usure créées par un contact répété entre objets. Eric Paquette [72] propose une méthode de génération d'impacts sur des objets maillés en leur appliquant des pressions à l'aide d'un outil virtuel dont la forme va déterminer la silhouette de l'impact. Le maillage de l'objet subit des raffinements locaux pour permettre la modification locale de la géométrie. Cette méthode permet de déformer localement et interactivement le maillage des objets. Cette méthode ne modifie pas la topologie des objets et ne permet que de légères déformations de la surface. La méthode proposée par Paquette est interactive ce qui permet au designer d'ajouter les impacts comme il le souhaite.

2.3.1.6 Les écaillures

Les travaux portant sur les écaillures traitent principalement du vieillissement des objets peints. Gobron et Chiba [39] ont développé une méthode *ad-hoc* pour créer des écaillures sur une couche de peinture mais aussi sur des objets en céramique. Leur approche consiste à utiliser les automates cellulaires afin de simuler la formation et la propagation des écaillures sur la surface. Les résultats obtenus par cette approche ne sont pas très réalistes.

Le modèle proposé par Paquette et *al.* [73] repose sur les lois de la physique. Ce modèle est basé sur un équilibre entre des forces d'élasticité et d'inélasticité aux seins de la couche de peinture. L'initialisation des forces est réalisée par une texture qui contrôle les zones de fortes et faibles tensions. Les résultats obtenus par cette méthode sont réalistes. Toutefois, la méthode d'initialisation ne permet pas de tenir compte d'événements environnementaux comme la présence d'une source de chaleur.

2.3.2 Fissures et Fractures

Les fissures et les fractures représentent des changements topologiques importants des objets (Figure 2.7 et 2.8). De plus, elles témoignent de son histoire. De ce fait, leur présence a un fort impact sur le réalisme des scènes de synthèses. Leur apparition est le résultat de forces de tensions qui conduisent à la dislocation de la matière. De nombreux travaux traitent de leur formation, notamment dans le domaine de la physique des matériaux. Dans cet état de l'art, nous nous intéressons plus particulièrement aux travaux menés dans le domaine de l'animation et de la modélisation physiquement réalistes. Ces travaux utilisent principalement une approche par simulation physique.

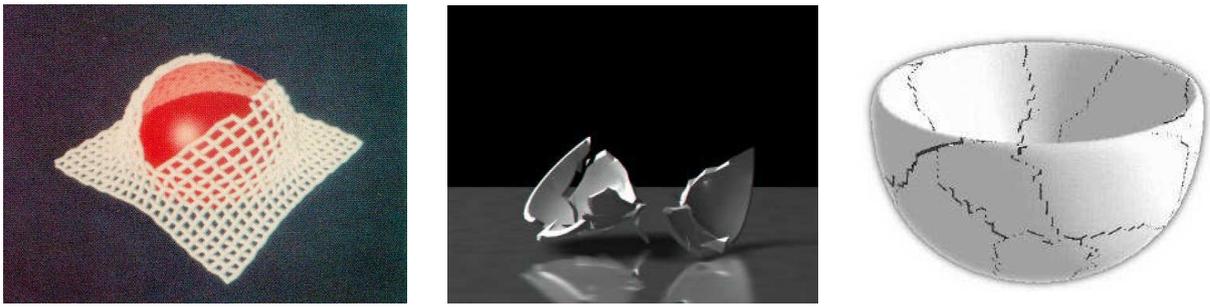


FIG. 2.7 – A gauche : L’une des premières simulation de l’animation de la déchirure d’un tissu réalisé par différence finies [44]. Au milieu : Simulation de la désagrégation d’un bol lors de sa chute sur le sol par éléments finis [71]. A droite : Simulation de la formation de fissures lors de l’assèchement d’un bol en argile par un système masses-ressorts [44]

2.3.2.1 Approches par différences finies

Terzopoulos et *al.* ont été des pionniers dans le domaine de l’animation et de la modélisation guidées par les lois de la physique. Dans [87, 86] ils proposent un modèle pour représenter les comportements inélastiques que sont : la viscosité, la plasticité et les phénomènes de fractures. Ce modèle repose sur la méthode des différences finies qui nécessite une discrétisation régulière des objets. Terzopoulos et *al.* réalisent l’intégration dans le temps par le biais de techniques adaptées à la taille du problème (Cholesky, gradient conjugué). Le modèle est appliqué à la simulation de la déchirure d’une feuille de papier et d’un morceau de tissu sous l’action d’une force extérieure. Cette approche donne des résultats physiquement réalistes. Toutefois, dans ce modèle, les fissures n’apparaissent qu’entre les éléments du maillage ce qui produit l’apparition de crénelages. De plus, les fissures ne peuvent pas se propager à l’intérieur du volume car l’objet est représenté par une grille régulière en deux dimensions.

2.3.2.2 Approches par éléments finis

Outre le fait que la méthode des différences finies ne permet pas de garantir la conservation de la masse, son principal inconvénient est qu’elle ne peut être employée que sur des formes simples. Afin de résoudre ces deux problèmes fondamentaux, plusieurs travaux ont été orientés vers l’utilisation de la méthode des éléments finis.

Dans [71] O’Brien et Hodgins proposent une méthode pour modéliser les fractures. Ils utilisent les lois de la mécanique des milieux continus pour exprimer la déformation du volume par des tenseurs de déformations. L’objet est représenté par un maillage de tétraèdres sur lequel est calculé un plan de fractures à partir des forces de tensions subies par le modèle. Un algorithme de remaillage local est utilisé pour découper les tétraèdres qui intersectent ce plan. Cette méthode permet de réduire considérablement les discontinuités liées à la discrétisation de l’objet. Toutefois, à chaque fois qu’un tétraèdre est subdivisé, il est nécessaire de calculer les paramètres des nœuds ajoutés. O’Brien et Hodgins utilisent leur méthode pour simuler

la formation de fractures sur un bol qui tombe au sol sous son propre poids, un mur heurté par un boulet, ou encore une plaque de verre brisée par un poids. Les résultats produits par cette approche sont très réalistes mais ils nécessitent des temps de calcul importants. Dans [70], O'Brien et *al.* étendent ce modèle au cas des matériaux ductiles. Là encore, les résultats obtenus sont très réalistes mais nécessitent de long temps de calculs.

Federl et Prusinkiewicz [35] utilisent la méthode des éléments finis pour simuler la formation de fissures dans des matériaux formés de deux couches. Leur modèle repose sur un maillage de prismes, attaché à la surface de l'objet. Les fissures sont produites en retirant les éléments dont le stress est supérieur au seuil du matériau. Les éléments situés le long du plan de la fissure sont subdivisés de manière dynamique de sorte qu'ils soient suffisamment petits par rapport à sa largeur pour ne pas produire d'artéfacts. Cette méthode produit des résultats réalistes mais ne peut être utilisée que sur des objets cylindriques ou plans.

Les résultats obtenus par les méthodes à base d'éléments finis sont physiquement réalistes. Toutefois, cette qualité nécessite des temps de calculs élevés. Par exemple, avec la méthode proposée par O'Brien et *al.*, il faut environ 28 minutes de calcul pour une seconde de simulation.

2.3.2.3 Approches par système masses-ressorts

Une autre approche est communément utilisée dans le domaine de la modélisation des fissures, c'est celle qui utilise les systèmes masses-ressorts. Ces systèmes sont relativement simples à mettre en œuvre et sont moins coûteux en termes de temps de calcul que l'approche par éléments finis.

Norton et *al.* [69] ont proposé une technique pour simuler la formation de fissures sur des objets subissant de fortes tensions. Dans leur modèle, l'objet est représenté par un maillage de cubes, déformé pour s'adapter au contour de l'objet. Ce maillage est directement relié à un système masses-ressorts. La formation des fissures est réalisée en supprimant les ressorts qui atteignent leur limite d'élasticité. Afin d'éviter la création de morceaux non consistants, l'intégralité du cube de ressorts est supprimée à chaque fois. De ce fait, et parce que le maillage cubique est très grossier, les fissures produites par cette approche présentent un très fort crénelage.

Hirota et *al.* [44] utilisent un système masses-ressorts pour simuler la formation des fissures qui apparaissent lors du processus d'assèchement d'objets en terre. Dans leur modèle, le système masses-ressorts est attaché à la surface de l'objet. L'apparition d'une fissure est modélisée par la rupture d'un ressort du réseau lorsque la tension qu'il subit est supérieure à son seuil d'élasticité. Ce modèle donne des résultats intéressants mais l'aspect des fissures reste peu réaliste du fait d'un crénelage important. De plus, ce modèle ne permet pas de représenter la propagation des fissures à l'intérieur du volume. Dans [45], Hirota et *al.* étendent leur méthode pour qu'il permette de représenter la propagation des fissures dans le volume de l'objet. Pour cela, ils associent un réseau de ressorts au maillage tétraédrique de l'objet. Comme dans leur modèle précédent, la fissure est générée en découpant les tétraèdres. Hirota et *al.* utilisent

leur modèle pour simuler la formation de fissures lors de l'assèchement d'un bloc d'argile. Les résultats qu'ils obtiennent sont réalistes d'un point de vue physique malgré une inconsistance temporelle. Cependant, les fissures n'ont pas un aspect très réaliste car elles sont extrêmement bruitées.

Les modèles masses-ressorts utilisent les principes de la physique, ce qui leur permet de produire des résultats physiquement réalistes. Toutefois, si ces modèles sont particulièrement bien adaptés à la simulation de petites déformations, ils présentent plusieurs limites dans les cas susceptibles de générer des fissures. La première limite est que, lorsque les contraintes sont trop élevées, il devient difficile d'estimer correctement la localisation et la direction de propagation de la fissure. Une autre limitation est due à la méthode employée pour modifier la topologie de l'objet. Lorsque la limite d'élasticité des ressorts est atteinte, ceux-ci sont retirés du réseau. De ce fait, la direction de la fissure suit la structure du réseau ce qui conduit à l'apparition d'artéfacts le long du front de propagation.

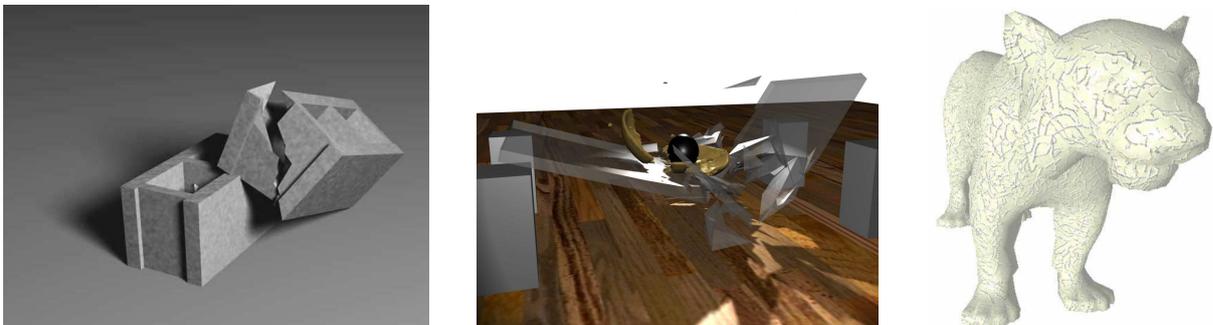


FIG. 2.8 – A gauche : Simulation en temps réel de la fracture d'objets rigides [63]. Au milieu : Simulation de la fracture d'objets fragiles [82]. A droite : Fissures générées en utilisant un automate cellulaire [41].

2.3.2.4 Approches hybrides

Quelques travaux ont été menés avec pour objectif l'obtention des résultats réalistes, obtenus avec les modèles à base d'éléments finis, associés aux faibles temps de calcul des modèles utilisant les systèmes Masses-Ressorts.

Smith et *al.* [82] proposent une méthode de simulation de fractures d'objets fragiles. Dans leur modèle, l'objet est représenté par un ensemble de masses interconnectées structuré par un maillage de tétraèdres. Contrairement aux systèmes masses-ressorts classiques, ce système repose sur la définition de contraintes de préservation de distance entre les masses. Cette approche leur permet de définir un système linéaire qu'ils résolvent par la méthode de descente de gradients de type gradients conjugués. En réduisant l'expression des contraintes à la préservation des distances entre les masses constituant l'objet, Smith et *al.* sont en mesure d'accélérer considérablement la simulation de la formation de fractures. Toutefois, cette approche n'est valable que pour des objets fragiles. De plus, étant donné qu'avec cette méthode les fissures

ne peuvent apparaître qu'entre les tétraèdres du maillage, ces dernières sont particulièrement bruitées.

Dans [63] Müller et *al.* présentent un système hybride pour la simulation de la déformation et de la fracture de matériaux en temps réel. Dans ce système, les objets sont représentés par un maillage de tétraèdres. Cette méthode reprend les principes proposés par O'Brien et *al.* dans [71] mais intègre deux simplifications nécessaires pour atteindre le temps réel. Tout d'abord, l'évaluation des forces d'élasticité n'est réalisée que lors des collisions. En dehors de ces instants, les éléments sont considérés comme étant rigides. Ensuite, Müller et *al.* partent du constat que les fissures dans des matériaux homogènes et isotropes ont tendance à être planes. Ils utilisent cette approximation pour découper les tétraèdres selon un plan de coupe orienté en fonction de la valeur du stress dans le matériau. Cette méthode ne donne pas des résultats très réalistes. De plus, malgré l'utilisation d'un plan de coupe, les fissures présentent un fort crénelage.

Molino et *al.* [62] introduisent l'algorithme de nœuds virtuels. Le principe de ce modèle repose sur la duplication du maillage tétraédrique du volume pour y stocker les zones dans lesquelles il n'y a plus de matière. Conjointement, l'algorithme génère la surface englobant pour chaque morceau de matière. Celle-ci est utilisée pour accélérer les calculs de collisions. La résolution du système d'équations produites par le modèle est réalisée en utilisant la méthode des éléments finis. L'intérêt majeur de cette méthode est qu'elle permet de produire des fractures dont l'orientation est totalement indépendante de la structure de la discrétisation. Molino et *al.* utilisent leur modèle pour sculpter, découper et fracturer des objets. Leur système semble particulièrement bien adapté à la simulation de coupe. En revanche, les fractures produites par leur système manquent de réalisme, notamment dans le cas de matériaux ductiles.

La plupart des méthodes précédentes modélisent la surface des fissures par les faces des éléments de la discrétisation, le plus souvent des tétraèdres. Cette approche nécessite le recours à une étape de raffinement local afin que la surface de la fracture ne paraisse pas irréaliste du fait d'un maillage trop grossier. Cette étape a coûté de calcul non négligeable et nécessite de recalculer les propriétés physiques des nouveaux nœuds qui rentrent alors en jeu dans la simulation. Pour résoudre ce problème, Pauly et *al.* ont développé un modèle permettant de simuler le comportement d'objets plastiques et élastiques susceptibles de se fracturer, et ce sans l'utilisation d'un maillage. Contrairement au cas des éléments finis où l'objet est discrétisé par un ensemble de formes finies, dans leur système, l'objet est représenté par un ensemble de points auxquels sont associés des fonctions de formes. La modélisation des fractures est réalisée en ajoutant à ces fonctions un terme représentant la discontinuité lorsqu'elles se trouvent le long du plan de propagation. Durant la propagation de la fissure, le volume de l'objet est échantillonné de manière adaptative. L'apport majeur de cette approche réside dans le découplage entre le domaine de la simulation et la représentation du contour de la surface de l'objet. Les fissures produites par cette méthode sont très réalistes. De plus, le temps de calcul est de l'ordre d'une vingtaine de secondes par pas de simulation ce qui est largement inférieur à une méthode par éléments finis.

2.3.2.5 Modèles spécifiques

D'autres approches ont été développées pour répondre à des problèmes très spécifiques. Par exemple, Gobron [41] utilise le concept d'automate cellulaire 3D pour simuler la création et la propagation de fissures dans différents matériaux tels que la céramique et la terre. La simulation tient compte du stress de la surface et de son relâchement lors de la création des craquelures. La position initiale des fissures est estimée à partir des informations de courbure locale et d'une carte d'épaisseur. Cette méthode donne des résultats satisfaisants mais nécessite la manipulation de paramètres peu intuitifs. De même, les temps de calcul sont de l'ordre de plusieurs heures.

Wyvill et *al.* [94] ont développé une méthode pour simuler la formation de craquelures qui se forment dans les batiks lors de la phase de séchage. Leur approche repose sur la notion de champ de distances. A partir d'une image initiale qui définit où a été versée la cire chaude, ils génèrent un champ de distances initial. De plus, ils constatent que les fissures vérifient les deux lois suivantes. Tout d'abord, la fissure se dirige toujours le plus rapidement possible vers les singularités de la surface comme une autre fissure, ou le bord de la cire. Ensuite, les nouvelles fissures sont situées dans des zones éloignées des vieilles craquelures. L'algorithme génère alors de nouvelles fissures conformément à ces deux règles et aux données du champ de distances. Cette approche produit des craquelures très réalistes mais reste assez spécifique. Elle offre toutefois un bon contrôle de la génération de fissures et est rapide.

2.3.2.6 Conclusion

On retrouve ici le même constat que celui fait pour la modélisation des détails des végétaux, à savoir que les détails microscopiques sont principalement traités par un plaquage de textures, quelque fois associé à une carte de BRDF pour rendre compte de la complexité de son interaction avec la lumière. C'est le cas par exemple des phénomènes d'oxydation, ou encore des rayures. D'autre part, les détails macroscopiques sont, quant à eux, généralement traités comme des éléments géométriques. C'est notamment le cas des fissures et des fractures qui produisent des changements topologiques importants de l'objet. De plus, on s'aperçoit que les textures sont principalement produites par une approche procédurale, tandis que la modification de la géométrie des objets est généralement le résultat d'une simulation physique.

Les simulations physiques produisent souvent des résultats très réalistes mais nécessitent des temps de calculs importants. Elles sont contrôlées par des paramètres physiques. Ainsi, dès lors que l'on connaît ces données, il est relativement simple de paramétrer la simulation. Or, c'est ici que réside aussi leur principale faiblesse, ces paramètres n'étant pas toujours faciles à connaître. A contrario, l'approche procédurale n'utilise pas forcément des paramètres physiques. Toutefois, en principe ces paramètres sont intuitifs et permettent un bon contrôle du résultat qui est souvent produit en temps réel, voire en temps interactif dans le pire des cas. Enfin, même si l'approche procédurale ne repose pas sur une simulation physique, elle peut permettre de générer un résultat physiquement plausible.

Afin de tirer le meilleur parti de chacune de ces approches, nous avons décidé d'utiliser une approche physique pour simuler les données de l'environnement telles que la lumière directe, la lumière indirecte, ou encore l'écoulement d'eau. Ces données étant importantes pour obtenir par la suite des résultats plausibles, il est raisonnable de prendre le temps de les calculer à partir des données de la scène. Toutefois, afin que ces paramètres n'aient pas à être recalculés, nous ferons l'hypothèse que l'ajout de détails ne modifie pas les données de l'environnement. Cette hypothèse peut être considérée comme vraie pour le cas de petits détails. A contrario, nous utiliserons une approche procédurale pour définir l'algorithme d'évolution dont le rôle est de combiner les éléments de l'atlas. En tenant compte de paramètres de l'environnement calculé précédemment, nous obtiendrons ainsi des résultats physiquement plausibles, et ce en temps interactifs. Ce dernier point est important car nous souhaitons offrir à l'utilisateur la possibilité de modifier le résultat.

2.4 Synthèse

Nous avons vu dans ce chapitre que la modélisation d'une scène complexe telle qu'une scène naturelle est généralement réalisée en deux étapes. On modélise d'abord le terrain et les objets de la scène puis on ajoute les détails. Deux approches sont principalement utilisées. La première approche consiste à générer le détail dans l'espace de la texture. A contrario, certaines méthodes tentent de modifier directement la géométrie de l'objet.

La grande majorité des méthodes de vieillissement rentre dans la première catégorie. Ainsi, l'ajout d'une tâche de rouille sur un objet en fer est traité de la même manière que l'ajout de mousse sur un objet en pierre : par la création d'une texture. Pourtant, si dans le cas d'une vue macroscopique, on peut faire l'hypothèse que la rouille n'entraîne pas de modification géométrique, ce n'est pas le cas de la mousse. Cet organisme va se développer et s'épaissir. Dans ce cas, un plaquage de textures ne sera plus suffisant pour représenter la richesse du détail. Par exemple, un plaquage de textures ne permet pas de gérer les effets d'auto-ombrage dans le cas d'une vue rapprochée.

Afin d'augmenter le réalisme des scènes virtuelles, il est donc intéressant de s'orienter vers l'ajout de détails géométriques. En général, les quelques travaux qui traitent de ce problème modifient directement la géométrie de l'objet. C'est par exemple le cas des écorces ou encore des fissures. Le principal problème de cette approche est que le fait de raffiner le maillage de l'objet pour ajouter du détail représente un coût mémoire important. D'autres méthodes utilisent l'objet initial comme un support. Les détails sont alors ajoutés sur la surface par des algorithmes de dispersion. C'est ainsi que sont traités les poils et les épines dans la littérature. Cette approche permet de minimiser les coûts en mémoire en dupliquant les objets. Toutefois, le problème de cette approche est qu'une répétition apparaît très vite du fait de la faible diversité des éléments dispersés. De plus, les éléments qui servent d'échantillons sont souvent de forme très simple.

Ainsi, de nombreuses techniques ont été développées mais aucune ne permet d'ajouter un

très grand nombre de détails dans une scène de synthèse tout en vérifiant nos objectifs qui sont :

- Création de détails réalistes même vue de près : puisque les détails sont partout dans la scène, ils se retrouvent forcément dans notre champ visuel, et ce à différentes échelles. Il est donc important de les modéliser précisément.
- Faible consommation mémoire : la gestion de la mémoire est un point très important notamment lorsque l'on veut pouvoir se déplacer dans la scène en temps réel.
- Création d'une grande variété de détails : Il est important de reproduire la diversité qui existe au sein d'une même famille de détails. C'est elle qui produit la richesse des tons que l'on retrouve dans la nature et c'est donc un acteur important vers le réalisme.
- Localisation réaliste : l'aspect réaliste des détails n'est pas le seul élément important, leur localisation l'est également. Si un détail est mal situé, il peut apporter des informations visuelles contradictoires qui vont nuire au réalisme de la scène.

Dans les chapitres qui suivent, nous nous attacherons tout particulièrement à remplir ces quatre objectifs.

Chapitre 3

Simulation de la propagation de lichen

Sommaire

3.1	Biologie du lichen	35
3.2	Dispersion du lichen	37
3.2.1	Dépôt manuel	38
3.2.2	Dépôt reposant sur l'accessibilité	38
3.3	Algorithme de propagation du lichen : l'Open-DLA	39
3.3.1	Le modèle du DLA	40
3.3.2	Algorithme de DLA intégrant la division cellulaire	41
3.3.3	Marche aléatoire	42
3.3.4	Détection de collision	43
3.3.5	Calcul de la probabilité d'agrégation	44
3.4	Atlas de lichen	47
3.4.1	Processus de substitution	48
3.4.2	Remplissage de l'Atlas	49
3.5	Synthèse	49

Publication Les travaux présentés dans ce chapitre ont fait l'objet d'une publication à la conférence Eurographics en 2003 sous le titre *Simulating and modeling lichen growth* [1].

Le lichen est un organisme biologique que l'on trouve partout dans la nature. Ils ont colonisé les pierres, les branches, les troncs mais aussi le sable et même certains métaux. Leur présence joue un rôle important dans le réalisme des scènes virtuelles. Elle donne des informations sur l'âge et l'orientation des arbres et des pierres d'un écosystème mais aussi des indications sur les conditions climatiques.

Dans ce chapitre, nous présentons un système pour modéliser et simuler la propagation du lichen dans une scène naturelle. Notre objectif est de générer des lichens réalistes, même vus à courte distance. Dans cette optique, un plaquage de texture n'est pas suffisant pour reproduire les formes complexes qu'ils exhibent dans la nature : une texture ne permettra pas d'obtenir les effets d'ombrage produit par certains lichens fruticuleux ou foliacés qui présentent de fortes ramifications.

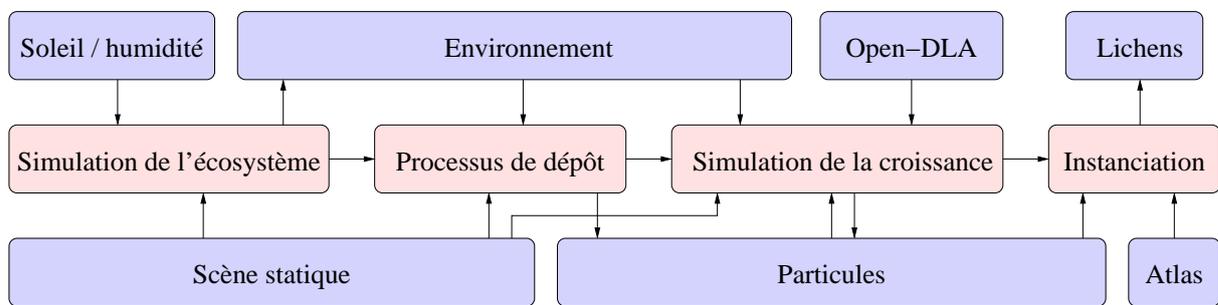


FIG. 3.1 – Aperçu du processus de simulation de l'ajout de lichen dans une scène virtuelle.

L'approche que nous proposons pour ajouter du lichen dans une scène naturelle se décompose en trois étapes (Figure 3.1). Tout d'abord, nous déposons des spores sur les objets de la scène. Ensuite, nous simulons leur développement. Pour cela, nous représentons les spores par un système de particules, lui-même contrôlé par le biais d'un algorithme de diffusion : l'Open DLA. Cet algorithme permet de tenir compte des caractéristiques de l'environnement, telles que la luminosité et l'humidité mais également des lichens se trouvant à proximité. Pour finir, au cours d'une étape d'instanciation, nous créons la géométrie de chaque lichen en remplaçant chaque particule par un élément de lichen en fonction de son âge. Cette étape repose sur un atlas de fragments de lichen.

La figure 3.2 illustre l'ensemble du processus dans le cas de la simulation de la colonisation d'une branche par les lichens de la famille des fruticuleux qui ont la particularité d'être fortement branchus.

3.1 Biologie du lichen

Les lichens sont des êtres doubles formés par l'association d'un champignon et d'un organisme contenant de la chlorophylle. Dans la majorité des cas, cet organisme est une algue verte microscopique mais il peut aussi s'agir d'une cyanobactérie.

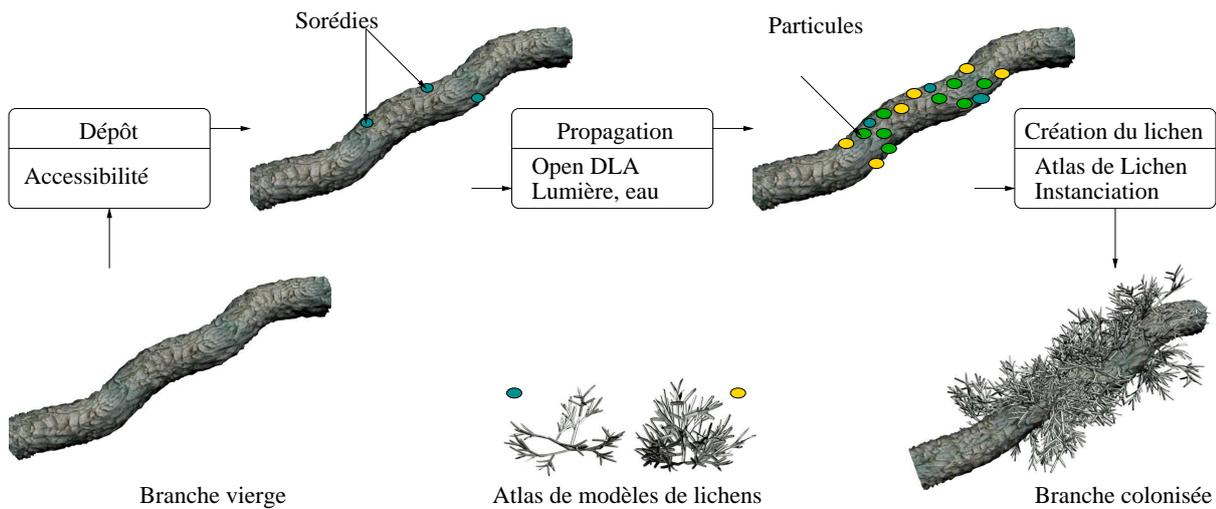


FIG. 3.2 – Principe de notre système de simulation de la croissance de lichen.

L’algue et le champignon forment la symbiose lichenique. Les deux partenaires tirent avantage de la présence de l’autre. Le champignon protège l’algue des rayons du soleil et lui fournit l’humidité dont elle a besoin pour ne pas se dessécher. De son côté, par le biais de la photosynthèse, l’algue fournit au champignon les molécules organiques nécessaires à son développement.



Caloplaca–Incospecta



Laboria–Linia



Hypogymnia–Heterophylla

FIG. 3.3 – A gauche : Lichen de la famille des crustacés. Au milieu : Lichen foliacés. A droite : Lichen branchu de la famille des fructiculeux.

Morphologie La forme du lichen est principalement définie par le champignon qui compose la majeure partie du thalle. Les lichens sont communément classés en trois catégories : les crustacés, les foliacés et les fructiculeux (Figure 3.3).

Les lichens crustacés sont aplatis et adhèrent fermement au substrat. On les retrouve prin-

cipalement sur les roches, le béton et les écorces des plantes. Les foliacés arborent des formes qui ressemblent à des feuilles. Ils sont faiblement rattachés au substrat. On les trouve sur les feuilles et les écorces des plantes. Les lichens fruticuleux sont souvent ramifiées et buissonnantes. Tout comme les foliacés, ils sont faiblement rattachés au substrat. On les trouve surtout sur l'écorce des plantes comme le bois.

Colonisation Les lichens peuvent se reproduire soit par reproduction sexuée, soit par multiplication végétative.

Dans le cas de la reproduction sexuée, les lichens émettent des spores par le biais de fructifications présentes à la surface du lichen. Ces spores sont transportés par le vent, les écoulements d'eau et les petits animaux. Ils doivent rencontrer un partenaire photosynthétique pour pouvoir poursuivre leur développement.

Les lichens se reproduisent principalement par multiplication végétative. Il s'agit d'une reproduction asexuée dans laquelle le lichen émet des sorédies, sortes de petites boutures. Celles-ci contiennent des cellules de champignons et d'algues. Les lichens sont ainsi capables de se disperser sur de grandes distances au gré du vent, des ruissellements et des animaux.

Croissance Les lichens sont capables de se développer sur les rochers, le sable, les sols nus, le bois, les fossiles, le béton et même sur les métaux. Constitués de l'association entre une algue et un champignon, ils peuvent coloniser des milieux particulièrement hostiles, telles que les régions désertiques mais aussi les régions arctiques. De plus, ils sont capables de végéter pour attendre des périodes favorables à leur développement. Durant cette période, leur thalle se dessèche et forme une croûte protégeant l'algue et l'empêchant de réaliser la photosynthèse. Le lichen vit alors au ralenti jusqu'à ce que les conditions d'humidité et de luminosité soient favorables. Cette propriété fait des lichens un pionnier exceptionnel et explique qu'ils soient extrêmement répandus.

La plupart des lichens, tels que ceux de la famille des crustacés, se développent très lentement, de l'ordre de quelques millimètres par an. Cette faible croissance est due au fait que l'algue qui réalise la photosynthèse est prisonnière du champignon et reçoit donc peu de lumière. A contrario, les lichens de la famille des foliacés se développent extrêmement rapidement avec une croissance de l'ordre de 20 millimètres par an. Chaque lichen a des besoins spécifiques. Toutefois, pour se développer dans de bonnes conditions, la majorité des lichens a besoin d'humidité et de lumière indirecte. La lumière directe brûle le thalle et provoque un dessèchement du lichen qui végète ou dépérit.

3.2 Dispersion du lichen

Les lichens se développent à partir des sorédies et produisent une nouvelle colonie s'ils se trouvent dans une zone qui regroupe de bonnes conditions de lumière et d'humidité. Le

processus de dépôt des sorédies est une étape importante de l'algorithme. C'est lui qui va influencer directement sur l'aspect réaliste de la répartition des lichens dans la scène. La simulation physique du transport des spores et des sorédies demanderait des calculs complexes et serait coûteuse en temps de calcul. Nous avons développé deux techniques pour contrôler leur dispersion dans la scène. La première technique consiste en un contrôle manuel. La seconde technique repose sur la valeur d'accessibilité de la surface.

3.2.1 Dépôt manuel

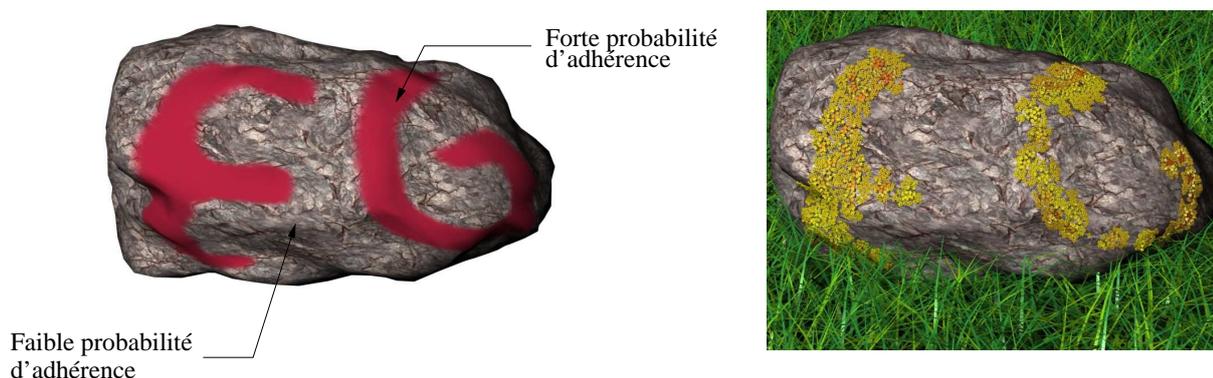


FIG. 3.4 – Résultat de la croissance (à droite) de sorédies dont la probabilité d'adhérence a été peinte directement sur la surface (à gauche).

Nous avons développé un outil simple permettant de définir, directement sur la surface, les zones dans lesquelles des spores peuvent se déposer. Par le biais d'un pinceau virtuel, l'utilisateur attribue à la surface une probabilité d'adhérence d'un spore. Pour cela, il peut faire varier l'opacité de la peinture. Lorsque l'opacité est maximum, la probabilité qu'un spore adhère est maximum et lorsque l'opacité est nulle, la probabilité est très faible. Nous dispersons ensuite aléatoirement des spores dans les régions définies par la peinture virtuelle. Les spores sont conservés lorsqu'ils vérifient la probabilité d'adhérence de leur région.

Sur la figure 3.4 nous avons utilisé cette technique pour contraindre les spores à rester dans une zone formant le symbole *EG*. Sur cet exemple, plus la peinture est rouge et plus la probabilité qu'un spore adhère à la surface est élevée. L'image de droite donne l'exemple d'une croissance réalisée à partir des spores placés aléatoirement dans les zones rouges.

3.2.2 Dépôt reposant sur l'accessibilité

Le contrôle manuel offre une grande liberté mais le résultat final dépend considérablement des connaissances du designer. Nous avons développé une technique qui s'appuie sur la valeur d'accessibilité de la surface pour évaluer la probabilité de présence de sorédies.

Dans la nature, la plupart des lichens apparaissent sur des surfaces qui sont facilement accessibles mais protégées contre le vent et la pluie. Cette propriété provient directement du

mode de reproduction des lichens. Les spores et les sorédies produits par le thalle sont transportés par le vent et les écoulements. Ils finissent leur périple sur les surfaces suffisamment exposées mais qui n'offre pas une exposition directe. Pour reproduire ce phénomène, nous déposons de manière aléatoire des spores dans certaines régions de la scène. Ces régions sont définies à partir du type de lichen et de l'accessibilité de la surface. Nous avons considéré qu'il s'agissait d'un facteur représentatif sous l'hypothèse que sur une grande période de temps, la surface est exposée à des vents et de la pluie provenant d'un grand nombre de directions. Pour calculer l'accessibilité d'une surface, nous avons utilisé la définition de Miller [59]. Lorsqu'un spore se retrouve dans une zone de forte accessibilité, nous le retirons. Ceci permet de simuler le fait qu'un spore ne peut pas se déposer sur une zone fortement soumise au vent et aux intempéries. De même, les spores sont éliminés des zones inaccessibles. Dans la nature, même si un spore atteignait une telle zone, par exemple par le biais d'un insecte, celui-ci végéterait et aurait une forte chance de dépérir.

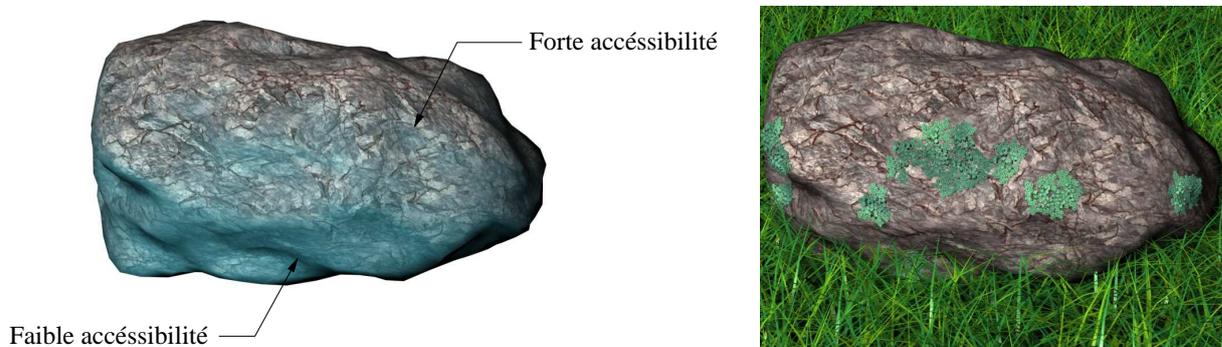


FIG. 3.5 – Colonie de lichens obtenue à partir des données d'accessibilité de la surface.

La figure 3.5 montre l'exemple d'une pierre colonisée par du lichen de la famille des crustacés. Ces lichens sont très résistants et peuvent se développer dans des zones très ensoleillées. Toutefois, les spores, eux, ne peuvent s'accrocher que dans des zones faiblement exposées au vent et aux averses. Les spores ont été déposées aléatoirement dans les zones présentant une accessibilité moyenne de l'ordre de 0.45 illustrée par la zone bleu clair.

3.3 Algorithme de propagation du lichen : l'Open-DLA

L'étude de la propagation du lichen montre qu'elle suit une loi de Pareto. La phrase *Rich get richer* caractérise bien cette loi. Cela signifie que plus on est riche et plus on s'enrichit. Dans le cas des lichens, cette loi traduit le fait que plus les longs lobes du lichen grandissent, plus de petits lobes se forment sur ceux-ci. Sumner [84] a montré que cette croissance peut être modélisée par un modèle de diffusion limitée par agrégation [92], noté DLA (Diffuse Limited Agregation). Ce modèle mathématique permet de générer des clusters de particules reproduisant des formes fractales constituées de longues dendrites.

Nous avons étendu ce modèle afin de mieux contrôler la forme produite et générer des formes caractéristiques pour une variété de lichen donnée.

Dans cette section, nous présentons notre modèle de croissance : l'open-DLA. Cet algorithme simule la propagation et la croissance de lichens en tenant compte des données de l'environnement, telles que la lumière directe, la lumière indirecte et l'humidité, mais aussi la présence d'autres lichens. Après avoir présenté notre algorithme de conquête, nous détaillerons chacun des points clés de la méthode qui sont la diffusion, le déplacement d'une particule, les tests de collision et enfin la définition de la probabilité d'agrégation.

3.3.1 Le modèle du DLA

Le principe du DLA est le suivant (figure 3.6). Tout d'abord, une nouvelle particule p_j est créée à partir d'une particule initiale dénotée p_0 . Ensuite, cette particule subit une marche aléatoire jusqu'à ce qu'elle entre en collision avec le cluster déjà formé. Si celle-ci vérifie la loi de probabilité d'agrégation $\mathcal{P}(p_j)$ alors elle est conservée et ajoutée au cluster. Sinon, elle est détruite. Le processus recommence alors de manière itérative.

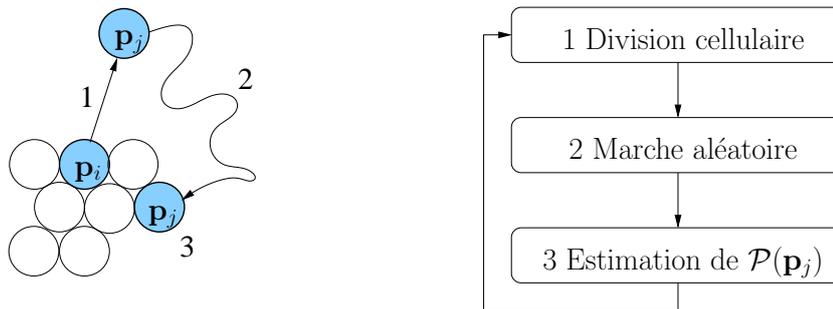


FIG. 3.6 – Principe de l'algorithme de diffusion limitée par agrégation.

L'algorithme se décompose en trois étapes (figure 3.6) : création d'une nouvelle particule, marche aléatoire sur la surface et enfin estimation de la probabilité d'agrégation de la particule.

Notons \mathcal{C} le cluster déjà formé et r le rayon des particules. Notre algorithme de propagation du lichen est le suivant :

1. Sélection aléatoire d'une particule p_i du cluster \mathcal{C} et création d'une nouvelle particule p_j à une distance $2r + \epsilon$ de celle-ci.
2. Déplacement de la particule p_j sur la surface par petits pas successifs et selon des directions aléatoires.
 - 2.1 Si p_j entre en contact avec le cluster \mathcal{C} alors, on teste si la probabilité d'agrégation $\mathcal{P}(p_j)$ est vérifiée. Si elle l'est, la particule est ajoutée au cluster, sinon elle est détruite.

2.2 Si p_j entre en collision avec un autre cluster ou se déplace trop loin de \mathcal{C} alors la particule est détruite.

Notre algorithme se distingue du modèle initial sur deux points : la phase de génération de nouvelles particules et la définition de la loi de probabilité d'agrégation $\mathcal{P}(p)$. Nous avons modifié la phase de génération de nouvelles particules pour qu'elle se rapproche davantage de ce que l'on trouve dans la nature : la division cellulaire. Enfin, nous avons défini la probabilité d'agrégation $\mathcal{P}(p)$ de sorte que la probabilité d'agrégation de la particule p intègre les caractéristiques de l'environnement.

3.3.2 Algorithme de DLA intégrant la division cellulaire

Dans les algorithmes classiques de diffusion limitée par agrégation, les nouvelles particules p_j sont créées à une distance R de la première particule p_0 du cluster \mathcal{C} . Ces particules suivent une direction initiale d_0 aléatoire. Afin d'éviter de produire des particules qui se déplaceraient indéfiniment, on fixe une distance seuil R_s de p_0 au delà de laquelle les particules p_i sont détruites. En général, R et R_s sont incrémentées en fonction de la taille du cluster.

Cette méthode est très simple à mettre en œuvre mais les particules peuvent réaliser un grand nombre de déplacements avant d'être en contact avec le cluster ou bien de dépasser la distance seuil : les temps de calcul sont donc extrêmement longs.

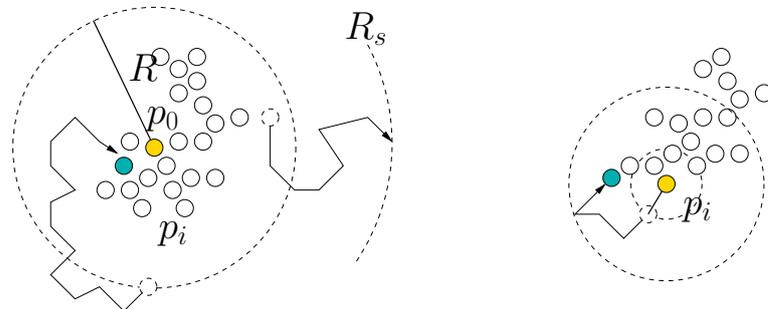


FIG. 3.7 – Processus de diffusion du DLA. A gauche : modèle originel. A droite : modèle simulant la division cellulaire.

Le modèle que nous avons développé permet d'augmenter l'efficacité de cette phase de création en reproduisant le phénomène de division cellulaire tel qu'on le trouve dans la nature. Pour mettre en place ce phénomène, nous avons modifié trois aspects de la phase de génération de nouvelles particules. Tout d'abord, la particule émettrice n'est plus nécessairement p_0 mais une particule p_i tirée aléatoirement parmi les particules du cluster. Ensuite, les nouvelles particules sont produites à une faible distance de \mathcal{C} . Si r est le rayon des particules, alors la nouvelle particule p_j est créée à une distance $2r + \epsilon$ du centre de p_i . Enfin, la particule p_j a comme direction initiale le vecteur $p_i p_j$. La figure 3.7 illustre la différence entre la méthode originale de création de nouvelles particules et notre approche.

Le paramètre ϵ permet de garantir que les particules sont créées suffisamment loin du cluster afin d'éviter que la particule émise p_j entre en contact immédiatement avec la particule p_i qui l'a générée. En pratique, nous utilisons $\epsilon = r/2$. Toutefois, dans un certain nombre de cas, la particule émise peut-être en collision avec d'autres particules du cluster au moment de sa création. Ceci est dû au fait que le choix de la particule émettrice est aléatoire et qu'il est donc possible qu'il n'y ai pas de place pour croître autour de celle-ci. C'est le cas par exemple lorsque p_i est située au centre d'un cluster très dense.

Afin de résoudre ce problème, nous avons mis en place un double système de jetons. Soit p_i une particule tirée aléatoirement, a_i et b_i les compteurs qui lui sont associés, alors le processus de génération de nouvelles particules est le suivant :

- 1 Si $b_i = 0$ alors la particule p_i est exclue du tirage aléatoire et on retourne à l'étape [1].
- 2 Si $b_i > 0$, on crée une nouvelle particule p_j à partir de p_i à une distance $2r + \epsilon$ dans une direction aléatoire.
 - 2.1 Si p_j est en contact avec le cluster alors on décrémente j_1 .
 - 2.2 Si $a_i = 0$ alors on décrémente b_i et on tire aléatoirement une nouvelle particule p_i du cluster puis on recommence le processus.
 - 2.2.1 Sinon on exécute l'étape de marche aléatoire.
 - 2.2.2 Si $a_i > 0$ on recommence l'étape [2].

Lorsque le compteur a_i est à zéro, une autre particule est tirée aléatoirement et le compteur b_i est décrémente d'une unité. Si le compteur a_i arrive à zéro alors la particule p_i ne peut plus émettre de nouvelle particule et sera définitivement exclue lors du tirage aléatoire.

Le fait d'utiliser deux compteurs plutôt qu'un seul permet de ne pas exclure une particule trop vite tout en poursuivant la croissance en cas d'échec avec celle-ci. La particule est définitivement exclue du tirage que lorsque l'échec se répète.

3.3.3 Marche aléatoire

La position d'une particule p est définie par l'index i du triangle T dans lequel elle se trouve et par son centre c .

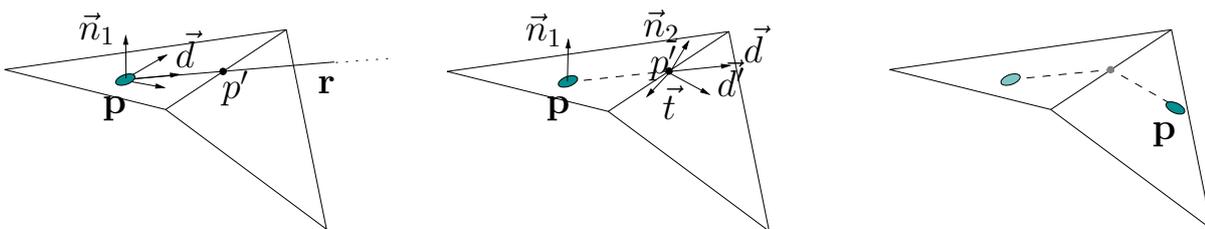


FIG. 3.8 – Algorithme de déplacement d'une particule sur une surface maillée.

Considérons une particule p de centre \mathbf{c} sur un triangle T_1 . Le processus de marche débute par la génération d'un vecteur \mathbf{d} donnant la direction de déplacement dans le repère local $(\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathbf{n}_1)$ de la particule. Lors du premier déplacement, ce vecteur est tiré aléatoirement dans le repère $(\mathbf{c}, \mathbf{x}, \mathbf{y})$ de la particule. Ensuite, ce vecteur est tiré aléatoirement sur un intervalle $[-\alpha, \alpha]$ autour de l'ancien vecteur de déplacement. L'angle α détermine la liberté qu'a la particule pour changer d'orientation. Le pas de déplacement, noté l , est fixé par l'utilisateur.

La nouvelle position de la particule est déterminée comme suit 3.8 :

1. Calcul de l'intersection \mathbf{p}' entre les arêtes du triangle T et le rayon r partant du centre de p avec comme vecteur unitaire \mathbf{d} .
2. Si $\|\mathbf{pp}'\| > l$ alors la nouvelle position de p est définie par $\mathbf{c} + l\mathbf{d}$
3. Sinon, la particule sort du triangle T . La particule est déplacée en \mathbf{p}' et l'index i devient l'index du nouveau triangle.

Soit \mathbf{n}_2 la normale du triangle T_2 partageant la même arête que T_1 , la nouvelle direction est calculée en évaluant le vecteur tangent $\mathbf{t} = \mathbf{d} \wedge \mathbf{n}_2$ ce qui permet de définir la nouvelle direction comme étant $\mathbf{d} = \mathbf{n}_2 \wedge \mathbf{t}$. La distance qui reste à parcourir est de $l = l - \|\mathbf{pp}'\|$. Enfin, on reprend à l'étape [1] de l'algorithme jusqu'à ce que l'on ait parcouru la distance initiale.

Lorsqu'une particule traverse deux faces parfaitement orthogonales, le vecteur tangent \mathbf{t} au déplacement ne peut être déterminé. Nous détectons ce cas en évaluant le produit scalaire $\mathbf{d} \cdot \mathbf{n}$. Si $\mathbf{d} \cdot \mathbf{n} \approx 1$ alors on perturbe légèrement \mathbf{d} avec le vecteur normale \mathbf{n}_1 de la face précédente.

3.3.4 Détection de collision

Les tests de collision entre les particules en mouvements et les clusters déjà formés constituent l'étape la plus gourmande en termes de temps de calcul. Tester chaque particule entre elles produit un algorithme de complexité en $O(n)$, ce qui entrainerait une croissance du cluster en $O(n^2)$.

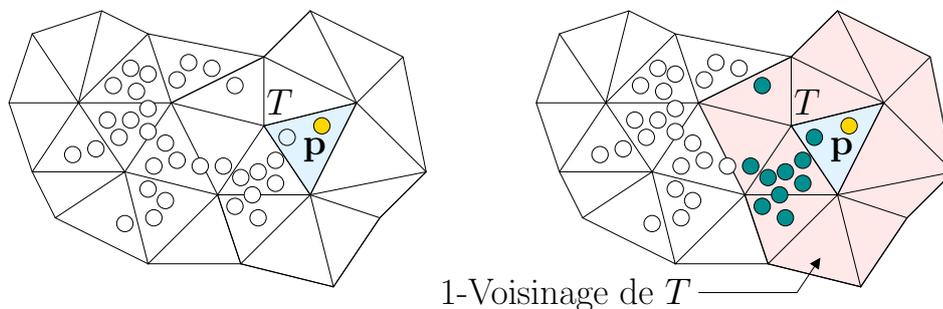


FIG. 3.9 – Détection de collision entre une particule (en jaune) et un sous-ensemble sélectionné dans son 1-voisinage (en bleu).

Nous créons pour chaque triangle du maillage une table contenant l'index des triangles qui sont dans son 1-voisinage. C'est à dire que chaque triangle possède l'index des triangles avec qui il partage une arête. Dans notre modèle, chaque particule p est attachée à un triangle T . Cela nous permet de ne tester que les collisions entre p et les particules attachées à T et son 1-voisinage. La figure 3.9 illustre ce principe. Dans notre système, le maillage des objets sur lesquels le lichen croît est stocké sous la forme de demi-arêtes. Cette structure nous permet de construire ces tables très rapidement lors d'une étape de prétraitement sur le maillage. Nous évitons ainsi le recours à des requêtes de voisinage durant la croissance, ce qui serait coûteux lors de la phase de colonisation.

Avec cette approche, la complexité de la détection de collisions entre une particule et un cluster de taille n est constante et n'est plus en $O(n)$.

3.3.5 Calcul de la probabilité d'agrégation

Nous définissons la probabilité d'agrégation d'une particule comme étant la combinaison de deux fonctions :

$$\mathcal{P}(p) = \mathcal{A}(p) \cdot \mathcal{E}(p)$$

$\mathcal{A}(p)$ est une fonction de diffusion qui définit pour une particule p sa probabilité d'agrégation théorique (dans des conditions idéales de croissance). Cette fonction caractérise la forme du cluster. La fonction $\mathcal{E}(p)$ définit quant à elle, pour une particule p , lorsque l'environnement lui est favorable et lorsqu'il ne l'est pas.

3.3.5.1 Contrôle de la forme du lichen

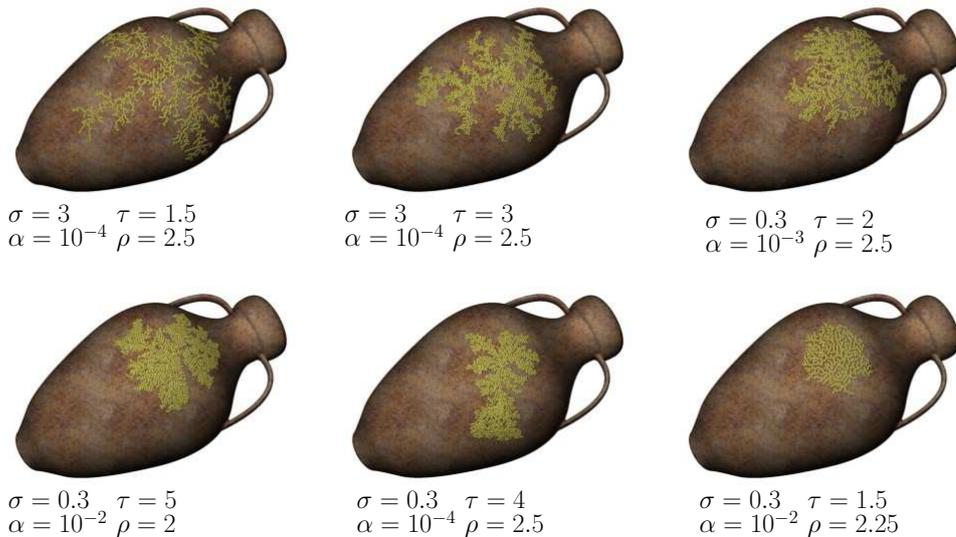


FIG. 3.10 – Exemple de différentes formes que nous obtenons en faisant varier les paramètres qui contrôlent la probabilité d'agrégation.

Nous contrôlons la forme du cluster par la fonction $\mathcal{A}(p)$. Cette fonction détermine la probabilité d'agrégation d'une particule p en fonction du nombre de particules dans son voisinage. Cette probabilité caractérise la forme du cluster indépendamment des caractéristiques de l'environnement et est définie par :

$$\mathcal{A}(p) = \alpha + (1 - \alpha)e^{-\sigma(n(p)-\tau)^2}$$

La fonction $n(p)$ désigne le nombre de voisins de la particule p . Nous évaluons cette quantité en sommant l'ensemble des particules qui sont à une distance ρ de p . Nous utilisons le paramètre ρ pour faire varier la densité du motif. Si r est le rayon d'une particule p , alors une petite valeur de ρ ($\rho \approx 2r$) produit des motifs compacts. A l'inverse, une valeur élevée ($\rho \geq 3r$) produit des motifs fortement dendritiques.

La figure 3.10 montre quelques exemples de lichens produits par notre système. Nous avons fait varier les paramètres de contrôle afin d'en illustrer l'impact sur la forme générée. Le paramètre σ contrôle la forme globale du cluster. Une valeur élevée de σ produit une forme très étendue tandis qu'une valeur très faible permet de générer une forme ramassée. Le paramètre τ caractérise l'aspect du cluster. Lorsque τ a une valeur élevée, le cluster est formé d'une multitude de lobes. A l'inverse, si τ est petit, le cluster est constitué de dendrites plus ou moins longues en fonction de σ . Enfin, le paramètre α permet d'introduire un biais dans la probabilité d'agrégation : il permet d'agréger des particules au cluster, même si celles-ci ne vérifient pas les propriétés de densité requises. Nous utilisons α pour augmenter la diversité d'aspect au sein du cluster. Nous évitons ainsi l'apparition de motifs trop uniformes qui seraient peu réalistes.

3.3.5.2 Influence de l'environnement

La fonction $\mathcal{E}(p)$ est définie par :

$$\mathcal{E}(p) = \min(\mathcal{I}(p), \mathcal{D}(p), \mathcal{H}(p))$$

Cette fonction estime pour un élément de lichen quelle est sa probabilité de se développer en fonction de sa sensibilité à la lumière directe, la lumière indirecte et l'humidité (Figure 3.11).

Nous définissons ses différentes sensibilités par trois courbes de réponse. Les courbes $\mathcal{I}(p)$, $\mathcal{D}(p)$, $\mathcal{H}(p)$ représentent respectivement, la sensibilité de la particule de lichen p à la lumière indirecte, la lumière directe et l'humidité. Nous évaluons la sensibilité de p en son centre. Ces fonctions retournent une valeur sur l'intervalle $[0,1]$. Une valeur proche de 1 signifie que les conditions sont favorables. Une valeur proche de 0 caractérise des conditions environnementales défavorables. Ainsi, $\mathcal{E}(p)$ est égale à 1 si les caractéristiques de l'environnement sont favorables en p . A contrario, si les caractéristiques de l'environnement sont défavorables à la croissance de p alors, la probabilité est de 0. Cela signifie qu'une particule p qui se trouve dans une zone défavorable (en rouge) a une forte chance de dépérir, et ce quelle que soit sa probabilité d'agrégation théorique. A contrario, une particule qui se trouve dans une zone favorable (en vert) a une forte chance d'être agrégée au cluster existant.

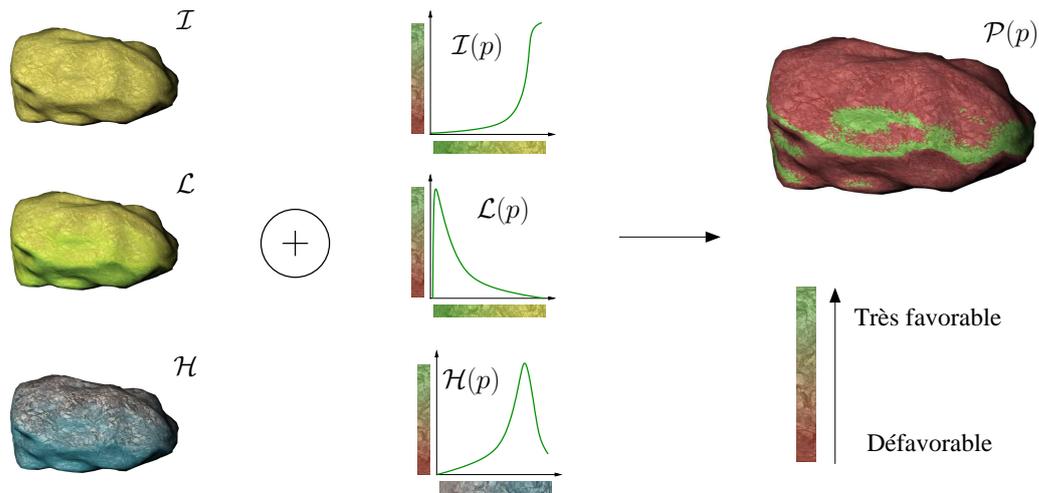


FIG. 3.11 – Processus de combinaison des caractéristiques de l’environnement et de la réponse du lichen.

Prendre en compte les caractéristiques de l’environnement est essentiel afin de produire des lichens aux formes réalistes ainsi qu’une distribution biologiquement plausible. L’étude de la biologie du lichen nous a permis de déterminer les facteurs environnementaux importants pour son développement. Nous avons ainsi constaté que la lumière indirecte et l’humidité ont un rôle favorable alors que la lumière directe est plutôt inhibitrice. Nous avons défini les caractéristiques de l’environnement par le biais de trois cartes : une carte pour la lumière directe, une carte pour la lumière indirecte et une carte pour l’humidité. Ces cartes donnent respectivement les quantités de lumière directe, de lumière indirecte et d’humidité sur la surface de la scène. Chacune de ces cartes peut être définie à la main ou bien être le résultat d’une simulation physique.

Les cartes de lumière directe et indirecte, notées \mathcal{D} et \mathcal{I} , peuvent être produites de manière précise par un calcul de radiosit  hi rarchique qui permet de simuler les  changes de lumi res entre les plantes et les objets de l’environnement [83]. Cette approche permet une simulation pr cise sur l’ensemble de la sc ne mais demande beaucoup de calculs. Dans notre syst me, nous n’avons besoin de g n rer ces cartes que l  o  le lichen va se d velopper. La phase de dispersion nous permet de conna tre les objets sur lesquels le lichen se trouve.

Pour calculer \mathcal{D} et \mathcal{I} pour un objet donn , nous avons besoin de d finir la trajectoire du soleil. Or, celui-ci suit une trajectoire complexe qui varie tout au long de l’ann e. Afin de d terminer sa contribution, nous avons fait l’hypoth se qu’il suit une trajectoire elliptique qui ne varie pas au cours du temps. Nous avons  galement utilis  un h misph re pour simuler la pr sence du ciel qui diffuse et renvoie une partie de la lumi re vers le sol. Nous utilisons une m thode de Mont  Carlo stochastique pour estimer la quantit  de lumi re directe et indirecte re ue en diff rents points de l’objet au cours d’une journ e. Nous  chantillonnons l’objet pour

lequel nous calculons ces cartes en subdivisant le maillage. Pour éviter de produire un effet de crénelage sur les cartes, nous calculons une nouvelle discrétisation de la trajectoire du soleil à chaque pas d'itération en utilisant la méthode proposée par Keller dans [48].

L'humidité joue un rôle considérable dans le processus de croissance du lichen. Ce sont des organismes très résistants, capables de se mettre en sommeil jusqu'à l'apparition de conditions d'humidité qui leur soient favorables. Par conséquent, cette carte est très importante. Toutefois, la simulation de l'écoulement de l'eau de pluie dans toute la scène et de l'influence des rivières est très coûteuse. Dans notre système, la carte d'humidité est définie par l'utilisateur. Cette carte est créée en peignant directement dans la scène les caractéristiques d'humidité et de sécheresse.

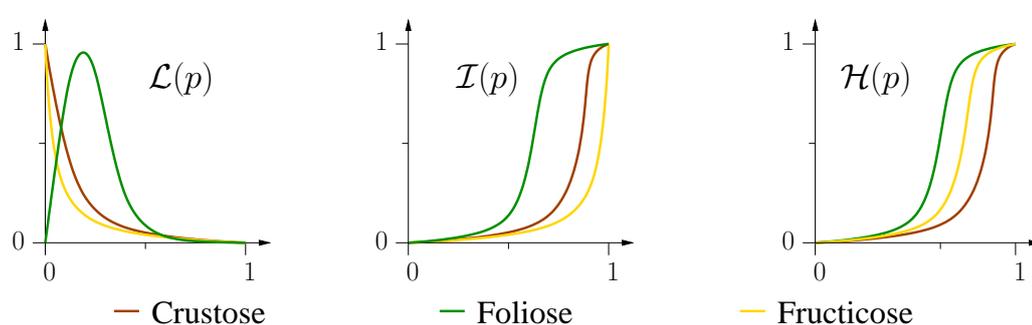


FIG. 3.12 – Courbes de sensibilité à la lumière et à l'humidité pour différentes variétés de lichens.

Chaque variété de lichen possède sa propre sensibilité à l'environnement. La figure 3.12 donne un exemple pour les trois grandes familles. Ces courbes ont été obtenues à partir d'observation faites sur un biotope dans lequel on retrouve ces lichens. Ces courbes traduisent le fait que les lichens de la famille des crustacés sont plus résistants à la lumière directe que les autres lichens. Les lichens de la famille des folioses ont, quant à eux, besoin de beaucoup d'humidité. Cela exprime le fait qu'on les trouve principalement sur des substrats poreux, tels que les écorces des arbres et arbustes.

3.4 Atlas de lichen

Les particules produites par l'algorithme de propagation sont remplacées une à une par un élément de lichen. Ces éléments sont organisés dans un atlas, lui-même organisé sous la forme de tables qui stockent, pour une variété de lichen, ses éléments caractéristiques classés en fonction de l'âge auquel ils peuvent apparaître (Figure 3.13). Ces fragments de lichens sont représentés par des modèles texturés. Chacun de ces modèles représente un élément caractéristique du lichen pour une famille donnée mais aussi à un âge donné. Il peut s'agir d'un fragment de thalle mais aussi d'un organe reproducteur. Chaque modèle \mathcal{M}_i d'une table possède un in-

tervalle $[a_i, b_i]$ qui définit sa probabilité d'apparition dans une table donnée. Si l'on note n le nombre d'éléments d'une table, ces intervalles vérifient les deux propriétés suivantes :

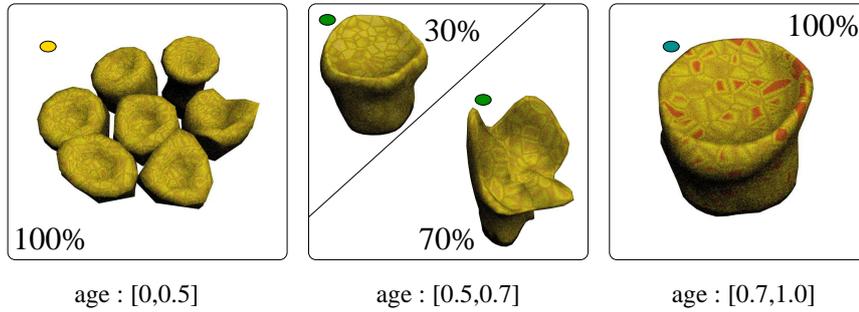


FIG. 3.13 – Exemple de fragments, classés en fonction de leur âge et de leur probabilité d'apparition, correspondant à l'atlas d'un lichen de la famille des crustacés.

$$[0, 1] = [a_0, b_0] \cup \dots \cup [a_i, b_i] \cup \dots \cup [a_n, b_n]$$

$$[a_0, b_0] \cap \dots \cap [a_i, b_i] \cap \dots \cap [a_n, b_n] = \emptyset$$

Cette classification par l'âge est un aspect important car elle permet de représenter la croissance du lichen par des étapes clés. Nous définissons la probabilité d'apparition de chaque fragment à partir d'observations faites dans la nature.

3.4.1 Processus de substitution

Après les phases de dispersion et de propagation, nous générons la géométrie de chaque lichen. Durant cette étape, les particules de chaque cluster sont remplacées par un modèle maillé de l'atlas en accord avec leur âge et le type de lichen.

Le processus de substitution est réalisé comme suit. Etant donné une particule p , nous sélectionnons dans l'atlas correspondant à la famille du lichen considéré, la table qui correspond à la tranche d'âge de p . Nous sélectionnons ensuite de manière aléatoire un modèle de cette table. Ce tirage est réalisé en accord avec la probabilité d'apparition des différents modèles de la table. Pour cela, nous tirons un nombre aléatoire n sur l'intervalle $[0,1]$ et nous sélectionnons le modèle \mathcal{M}_i tel que $n \in [a_i, b_i]$.

La figure 3.14 illustre ce processus de substitution pour du lichen de la famille des fruticuleux. A la fin de la propagation, nous avons obtenu un ensemble de clusters formés de particules de différents âges qui ont colonisé la branche. Les particules les plus jeunes se retrouvent sur le pourtour des clusters tandis que les particules les plus vieilles sont à l'intérieur. Les particules du pourtour sont remplacées par de petits modèles peu touffus. A contrario, les particules les plus vieilles sont remplacées par des modèles fortement ramifiés et denses.

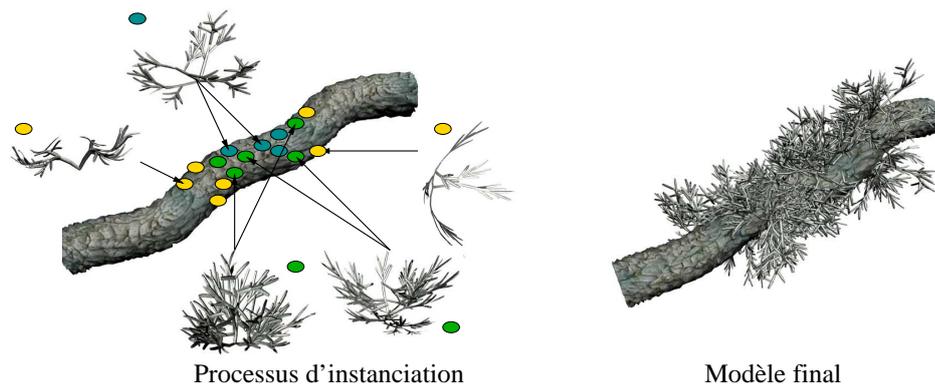


FIG. 3.14 – Processus de substitution des particules par des fragments de lichen.

3.4.2 Remplissage de l'Atlas

La création des éléments de lichens et leur organisation dans l'atlas joue un rôle important dans l'aspect réaliste des lichens produits par notre méthode. Nous avons développé des outils adaptés à la modélisation des fragments de lichens de la famille des crustacés ainsi que pour ceux de la famille des fructiculeux.

Les lichens crustacés forment une croûte de faible épaisseur sur la surface des objets. Nous modélisons leurs fragments par le biais d'une carte de hauteur. Nous produisons celle-ci soit en capturant le relief à partir d'une photographie, soit en la générant par une fonction de bruit. Les lichens crustacés peuvent également présenter des éléments qui se détachent nettement du sol comme des apothécies par exemple. Nous modélisons ces éléments en utilisant des cylindres généralisés.

Les lichens fructiculeux sont formés d'une multitude de petites branches fortement ramifiées. Nous avons choisi de modéliser leur fragment à l'aide de L-Systèmes qui sont bien adaptés pour représenter ce type de géométrie.

La texture associée aux fragments de lichens est obtenue à partir de photographies ou bien reproduite procéduralement. Dans le cas des lichens crustacés, la texture de la croûte est issue d'une photographie et celle des apothécies est générée par une texture cellulaire. Pour les lichens fructiculeux, la texture des branches est produite par une texture procédurale.

3.5 Synthèse

Notre modèle d'Open-DLA permet de modéliser des lichens relativement plats et d'aspect rugueux comme les lichens crustacés mais également, des lichens volumineux tels que les lichens fructiculeux. De plus, en prenant en compte les données de l'environnement durant le processus de colonisation, notre modèle produit des lichens dont la forme et la répartition sont plausibles selon des critères biologiques.

Nous avons simulé la propagation de lichen de la famille des crustacés sur le béton (Fi-

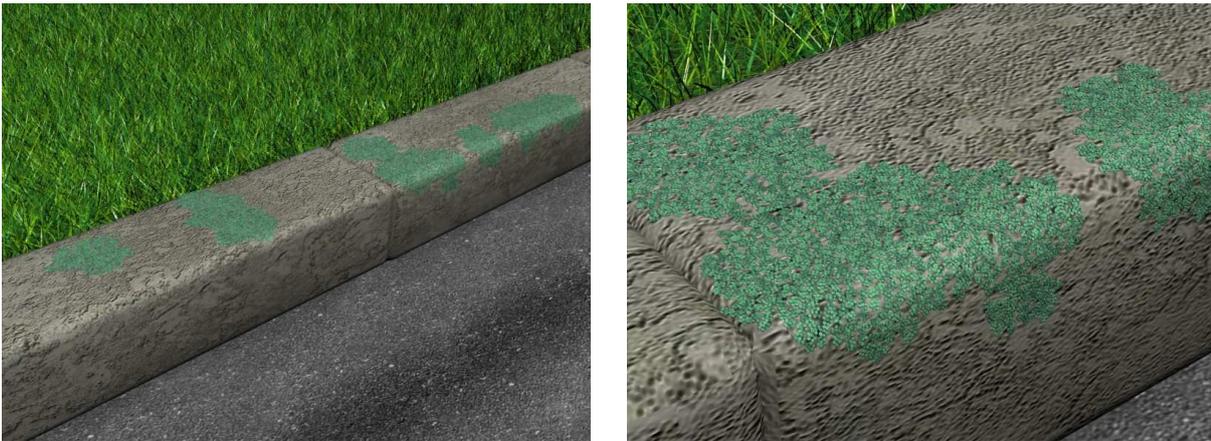


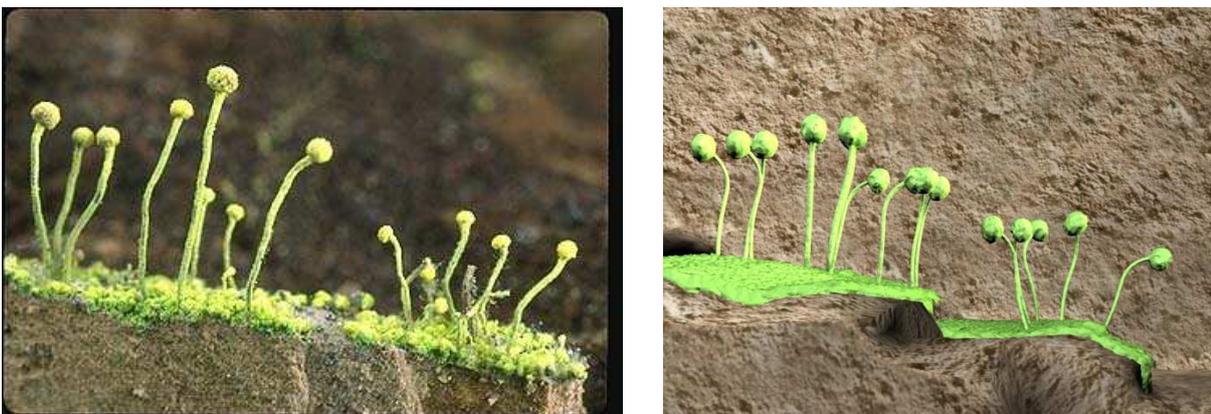
FIG. 3.15 – Lichens crustacés ayant colonisé un trottoir en béton.



Photographie

Lichen de synthèse

FIG. 3.16 – Comparaison entre une photographie et une simulation de la croissance du *Caloplaca inconspicua*.



Photographie

Lichen de synthèse

FIG. 3.17 – Comparaison entre une photographie et une simulation de la croissance du *Chenotheca furfuracea*.

gure 3.15 et 3.17) et la roche (Figure 3.16).



Photographie



Lichen de synthèse

FIG. 3.18 – Comparaison entre une photographie et une simulation de la croissance du *Hypogymnia heterophylla*.

La figure 3.18 illustre le résultat obtenu lors de la simulation du développement de lichens de la famille des fruticuleux : le *Hypogymnia heterophylla*. La simulation reproduit bien l'aspect général des lichens de cette famille. Toutefois, sur cet exemple, on constate la nécessité d'avoir des outils adaptés pour modéliser les fragments de lichens. Les modèles utilisés ici sont trop fins et dépourvus d'organes reproducteurs, ce qui nuit au réalisme de la forme finale. Le résultat final est suffisant pour une vue peu rapprochée, mais il n'est pas suffisamment détaillé pour une vue macroscopique.

Certains types de lichens ne peuvent pas être représentés par notre modèle. C'est le cas notamment des lichens qui forment de longs filaments qui s'enchevêtrent tels que l'*Alectoria-Sarmentosa* (Figure 3.22).



Scène initiale



Scène colonisée

FIG. 3.19 – Scène naturelle colonisée par des lichens de la famille des crustacés.



FIG. 3.20 – Exemple d’une scène complexe colonisée par différents types de lichens.



FIG. 3.21 – Mascotte d’Eurographics’2004 avant et après colonisation par des lichens de la famille des crustacés mais aussi des fructiculeux.

	Triangles	Clusters	Particules	Propagation
Pierre de la figure 3.20(lichen vert)	450	5	505	50s
Pierre de la figure 3.20(lichen jeune)	2400	71	2400	483s
Arbre de la figure 3.20	450	5	54	1s
Mascotte de la figure 3.21	8388	26	2000	150s
Branches de la figure 3.20	8212	24	522	26s

TAB. 3.1 – Temps nécessaire, en seconde, à la simulation de la propagation du lichen des différentes scènes en utilisant l’open-DLA.

Notre système permet de simuler la colonisation d'une scène par différents types de lichens. Lors de cette étape, un grand nombre de particules sont générées puis, substituées par des modèles maillés. La table 3.1 donne quelques statistiques liées à la simulation de la propagation de lichen dans le cas de trois scènes virtuelles (Figures 3.19, 3.20 et 3.21). Parmi ces informations, on retrouve : le temps de calcul de la simulation, le nombre de sorédies, le nombre de particules générées et le nombre de triangle du lichen avec et sans instanciation.

Dans la plupart des cas, le temps nécessaire à la propagation du lichen est inférieur à 1 minute. La seule exception est celle de la pierre de la scène de la figure 3.20 qui est recouverte de lichens crustacés. Ceci est dû au fait qu'un grand nombre de clusters en contact sont présents sur la pierre ce qui nécessite énormément de tests de collision.



FIG. 3.22 – A gauche : Photographie d'Alectoria-Sarmentosa. A droite : Colonie de lichens formée principalement par le phénomène de ruissellement de l'eau de pluie.

Enfin, dans notre système, la dispersion et la propagation du lichen peut-être réalisée automatiquement en calculant l'accessibilité de la surface, ou bien être définie par l'utilisateur. Le contrôle manuel est très intuitif puisqu'il s'agit de peindre directement sur la surface les régions de l'objet sur lesquelles chaque type de lichen peut se déposer et croître. La section 3.5 montre qu'avec cette approche, nous pouvons reproduire un grand nombre de lichen très facilement. Toutefois, celle-ci ne permet de reproduire des distributions comme celle illustrée par la figure 3.22. Pour reproduire ces formes dues à l'entraînement des sorédies par les eaux de ruissellement, il faudrait réaliser une simulation d'écoulement d'eau. Un modèle tel que celui proposé par Dorsey dans [33] serait adapté à ce problème. Toutefois, ce type de simulation est beaucoup plus coûteux en termes de temps de calcul mais est néanmoins possible avec cette approche.

Chapitre 4

Simulation de la chute et du vieillissement des feuilles

Sommaire

4.1	Phénomènes chimiques et biologiques et coloration des feuilles	58
4.2	Modélisation de l'atlas	59
4.3	Processus de vieillissement	61
4.3.1	Caractéristiques de l'environnement	61
4.3.2	L'Open Markov-Chain	62
4.4	Simulation de la chute de feuilles	64
4.4.1	Processus de détachement	65
4.4.2	Simulation de la chute des feuilles	65
4.4.3	Compactage des tas de feuilles	68
4.5	Résultat et discussion	68

Publication Les travaux présentés ici ont été publiés à la conférence Eurographics en 2006, sous le titre *Simulating autumn coloring and falling leaves* [5].

La modélisation du vieillissement des feuilles et la simulation de leur distribution dans une scène virtuelle est un problème fondamental en informatique graphique. Le processus de vieillissement des feuilles est très complexe car il n’inclut pas que le changement de couleur que l’on peut observer en automne mais aussi une déformation de la feuille lorsqu’elle sèche. On peut également constater un phénomène de putréfaction. De plus, l’environnement joue un rôle significatif dans la manière dont ce vieillissement va évoluer. Ainsi la durée du jour, la température et l’humidité sont des paramètres dont il faut tenir compte si l’on veut pouvoir reproduire la dégradation d’une feuille au cours du temps de manière réaliste.

Ensuite, la distribution des feuilles est le résultat d’un ensemble de phénomènes dynamiques. Elles volent, planent et tourbillonnent dans le vent mais aussi glissent et roulent sur le sol. La simulation de ces phénomènes est très complexe. Une simulation précise de ces déplacements serait très coûteuse en temps de calcul. De plus, la manière dont la feuille va interagir avec les flux venteux et son environnement est difficilement contrôlable, ce qui rend une simulation peu utile.

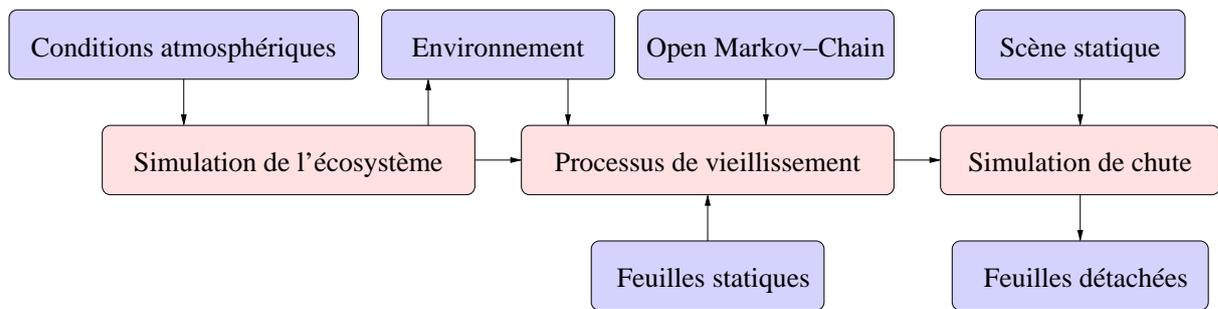


FIG. 4.1 – Aperçu du processus de simulation de vieillissement de feuilles.

Dans ce chapitre, nous présentons un système permettant de simuler le vieillissement et la distribution de feuilles dans une scène automnale. Nous avons développé un modèle qui permet de représenter le cycle de vie d’un ensemble de feuilles. Celui-ci est associé à un algorithme de chute qui permet de reproduire de manière procédurale une distribution physiquement plausible des feuilles sur le sol. Notre objectif n’est pas de vieillir et de déposer une feuille. Nous nous intéressons à l’évolution stochastique du feuillage d’un ou plusieurs arbres. L’objectif de ce travail est de reproduire une distribution réaliste des feuilles dans une scène virtuelle. Ce réalisme concerne d’une part la répartition en termes de teintes et de couleurs telle que l’on peut l’observer en automne, et d’autre part, la distribution des feuilles au sol qui peuvent former des tas par accumulation.

Nous simulons ces processus de vieillissement et de chute de feuilles en automne en tenant compte des caractéristiques de l’environnement. Notre système d’évolution est contrôlé par un modèle de vieillissement de feuilles : l’Open Markov Chain (Section 4.3). Ce modèle définit de manière statistique leur évolution en fonction des caractéristiques de l’environnement.

Le processus de simulation du vieillissement est constitué de trois étapes (Figure 4.1). Tout d’abord, nous évaluons les caractéristiques de l’environnement (Section 4.3.1). Nous générons

la carte de température qui joue un rôle important dans le processus de vieillissement. Nous calculons également une carte d'humidité qui a une influence sur la dégradation et la putréfaction des feuilles. Ensuite, nous appliquons notre modèle de l'Open Markov Chain aux feuilles de la scène. Pour chaque feuille, nous évaluons sa probabilité d'évolution en fonction des conditions de température et d'humidité locales.

Enfin, nous appliquons nos algorithmes de sélection, chute et tassement (Section 4.4) pour générer la distribution finale des feuilles dans la scène. Les feuilles sont détachées et chutent en accord avec les données de l'environnement.

La section 4.5 montre quelques résultats que nous avons obtenus par cette approche.

4.1 Phénomènes chimiques et biologiques et coloration des feuilles

La couleur automnale des feuilles est le résultat de la dégradation de trois pigments qui sont la chlorophylle, le carotène et l'anthocyane. La chlorophylle et le carotène, qui sont stockés dans les cellules des feuilles, donnent des couleurs allant du vert au jaune en fonction de leurs taux de concentration respectifs. Par contre, l'anthocyane, qui circule dans la sève des feuilles, est responsable de la coloration rouge de certaines feuilles, comme celles de l'érable. Lorsque la feuille meurt, ces pigments se détériorent et des tanins apparaissent (Figure 4.2). Ceux-ci sont responsables de la coloration brune des feuilles en automne.

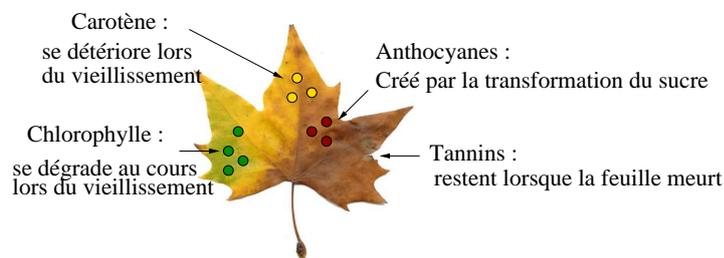


FIG. 4.2 – Molécules responsables de la coloration des feuilles.

Les conditions qui déclenchent la dégradation des pigments, et ainsi le changement de couleurs, sont la variation de température, la quantité de lumière qui décroît en automne et le taux d'humidité. La température est le principal facteur déclenchant. Lorsque celle-ci diminue durant la nuit, un bouchon de sève se forme entre la tige de la feuille et la branche qui la nourrit. Le sucre, qui est produit par la photosynthèse, s'accumule dans la feuille et est transformé en anthocyane. Si la sève est acide, l'anthocyane a une coloration rouge claire. A contrario, la coloration de l'anthocyane est rouge foncée, voir pourpre dans un milieu plus basique. Lorsque la température continue à baisser, la chlorophylle contenue dans la feuille va se dégrader, suivie par le carotène qui est plus résistant. C'est l'ensemble de ce processus de création d'anthocyane et de dégradation de la chlorophylle puis du carotène qui est responsable de la variation de couleur que l'on peut observer en automne (Figure 4.3).



FIG. 4.3 – Exemple de la richesse de la coloration des feuilles en automne.

Le taux de concentration de ces trois pigments est différent pour chaque variété d'arbre. Le carotène est très présent dans les feuilles du bouleau, du peuplier et du noyer. Par conséquent, ces arbres ont une coloration qui tend vers le jaune durant l'automne. En revanche, les arbres tels que l'érable, le chêne et le sumac vinaigrier, contiennent plus d'anthocyane et arborent alors des tons automnaux qui tendent vers le rouge.

4.2 Modélisation de l'atlas

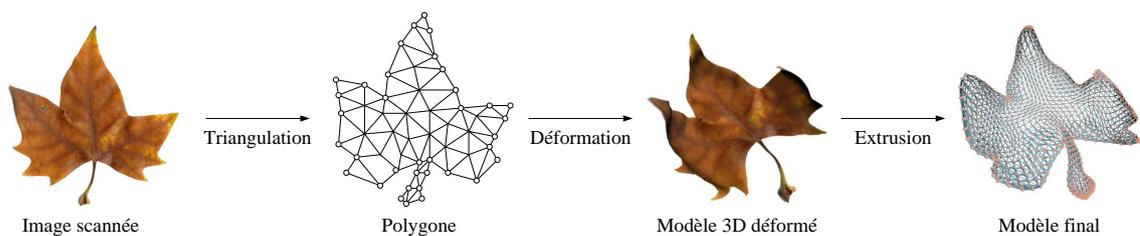


FIG. 4.4 – Processus de modélisation d'une feuille.

Les feuilles en automne arborent une grande variété de coloris, de textures et de formes. On peut ainsi observer des feuilles allant du vert au jaune, légèrement coudées, mais aussi des feuilles totalement desséchées, allant du jaune au brun et s'enroulant sur elles-mêmes. Modéliser toutes ces feuilles serait très compliqué et demanderait un vrai travail d'artiste. Notre approche consiste à modéliser quelques feuilles définissant un ensemble réduit de modèles et à les stocker dans un atlas.

L'atlas est organisé par paquets d'âges et de formes. Ceci nous permet de regrouper les modèles de feuilles qui peuvent être utilisés pour un âge donné. La figure 4.5 illustre de manière simplifiée la structure de l'atlas.

Les feuilles sont décrites par un maillage associé à une texture scannée. Dans la scène, chaque modèle est défini comme une instance afin de réduire leur coût en mémoire. Pour créer le modèle maillé et texturé d'une feuille, nous procédons en trois étapes (Figure 4.4). Tout d'abord, nous créons un maillage en deux dimensions de la feuille en extrayant son contour à partir du scan [64]. Le maillage est généré par une triangulation de Delaunay à partir du polygone ainsi obtenu. Cette méthode nous permet de générer un maillage dont la résolution est

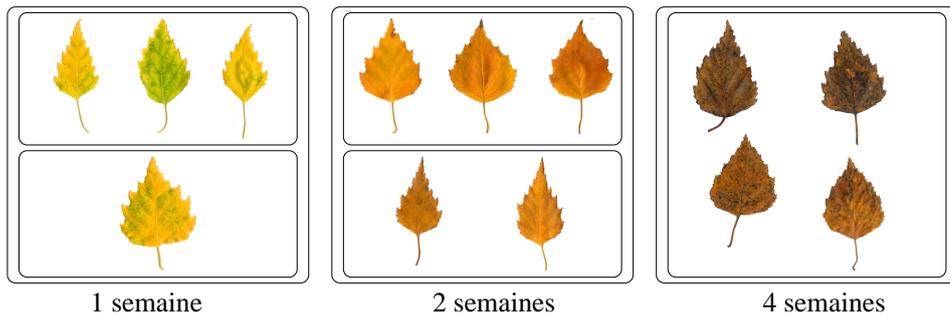


FIG. 4.5 – Schématisation simplifiée de l'atlas.

adaptée au niveau de détails souhaité. Ensuite, nous déformons ce maillage dans un espace à trois dimensions. Cette étape a pour objectif d'augmenter le réalisme du modèle en lui donnant du volume. Pour réaliser des déformations importantes, comme des plis ou des ondulations, nous utilisons un système Masses-Ressorts associé à la feuille. Ceci nous permet de maintenir une surface à peu près constante tout en appliquant des déformations importantes. Les petites déformations surfaciques sont créées en déplaçant les sommets du maillage le long de la normale par des valeurs aléatoires. Enfin, nous obtenons le modèle final en extrudant légèrement le maillage déformé. Cette étape a pour but de donner une épaisseur à la feuille ce qui nous permet alors de lui appliquer six textures (Figure 4.6) : trois pour la face avant et trois pour la face arrière.

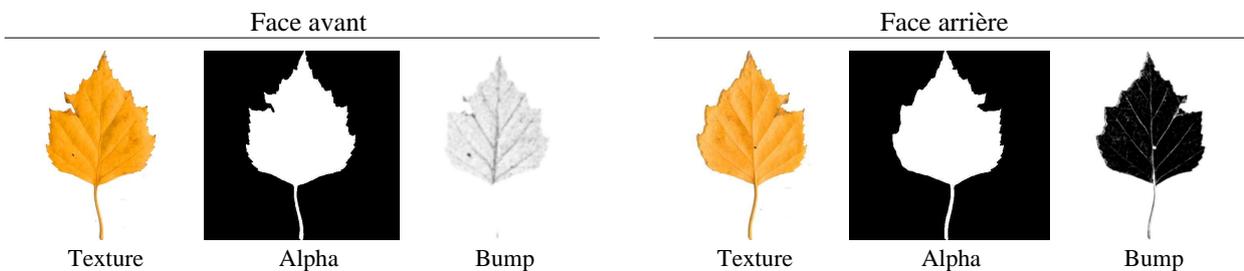


FIG. 4.6 – A gauche : Textures appliquées sur la face avant de la feuille. A droite : Textures appliquées à la face arrière de la feuille.

La texture de la face avant et de la face arrière est obtenue en scannant une feuille. Ensuite, le canal alpha et le bump sont générés automatiquement en appliquant des filtres classiques sur l'image : passage en niveau de gris, détection de contours et flou Gaussien.

Lors du processus de plaquage de textures, un recalage rigide peut être nécessaire afin de faire correspondre la face avant et la face arrière. Ceci est particulièrement visible lorsque la feuille est trouée. Dans ce cas, le trou est défini dans l'espace texture par le canal alpha de la face avant d'une part et de la face arrière d'autre part. Pour obtenir un modèle cohérent, il est nécessaire que les deux faces soient en correspondance.

Cette technique de modélisation nous permet de reproduire facilement la coloration très complexe des feuilles en automne. La figure 4.7 illustre le type de feuille que nous avons créé pour remplir l’atlas.

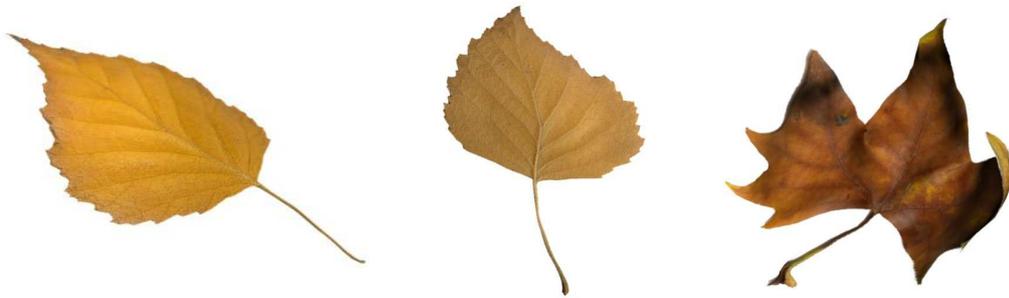


FIG. 4.7 – Exemple de modèle de feuilles texturées et déformées.

On peut constater que l’utilisation d’une texture scannée et la génération du bump et de l’alpha correspondant donne un très bon résultat. Les modèles ont des couleurs très réalistes. Toutefois, cette approche pose le problème de la déformation. Tordre, incurver et flétrir ces feuilles n’est pas une tâche aisée. Nous avons utilisé un système Masses-ressorts pour réaliser cette étape. L’utilisation de ce type de modèle ne permet pas de produire facilement des feuilles qui s’enroulent sur elle même. Le problème de la modélisation des instances de feuilles reste un problème ouvert. Malgré les nombreuses techniques proposées, le problème de la création de feuilles réalistes n’est pas résolu.

4.3 Processus de vieillissement

Dans la section 4.1, nous avons vu que la coloration des feuilles est le résultat de la dégradation de trois pigments. De plus, nous savons que cette dégradation est influencée par l’environnement. Après avoir présenté comment nous définissons l’environnement, nous allons décrire notre modèle qui repose sur les chaînes de Markov. Nous verrons alors comment celui-ci peut permettre de représenter le cycle de vieillissement des feuilles.

4.3.1 Caractéristiques de l’environnement

Tenir compte des caractéristiques de l’environnement est essentiel. Les conditions atmosphériques ont un impact direct sur la coloration des feuilles mais aussi sur leur répartition dans une scène naturelle. Par exemple, dans les zones humides, les feuilles seront majoritairement putréfiées et auront des tâches brunes. A l’inverse, dans des zones sèches, exposées au soleil, les feuilles seront plutôt jaunes et sans tâche.

Les travaux précédents privilégiaient soit la lumière du jour [61], soit la température [20] comme facteur déclenchant du changement de couleur. Or, dans la nature, ce phénomène est bien plus complexe car il dépend de ces deux facteurs qui sont corrélés. Dans notre système,

nous utilisons la température et l'humidité, notée respectivement t et w , comme facteurs déclenchants. Ces deux éléments jouent un rôle majeur dans le processus de putréfaction et de dégradation de la feuille.

Afin de simplifier les notations, dans ce chapitre, les caractéristiques de l'environnement seront définies par $\mathbf{e} = (t, w)$. De plus, t et w font référence à une échelle de température et d'humidité, c'est à dire que l'on a $t \in [0, 1]$ et $w \in [0, 1]$.

Température La température est liée à la lumière directe et indirecte du soleil qui chauffe les surfaces accessibles. Nous créons la carte de température, notée \mathcal{T} en combinant les cartes de lumière directe et de lumière indirecte. Nous évaluons la carte de lumière directe en intégrant la quantité de lumière émise par le soleil au cours d'une journée. Nous discrétisons la trajectoire du soleil en ensemble de sources ponctuelles en utilisant un échantillonnage stochastique [49]. Cette méthode nous permet d'éviter la présence de crénelage sur la carte qui nuirait à la simulation du vieillissement.

Humidité L'humidité locale de la surface pourrait être calculée en utilisant un système de particules contraint sur la surface comme présenté dans [33]. Cette méthode permet de suivre précisément le chemin emprunté par l'eau de pluie qui ruisselle sur les surfaces et produit des zones sèches et humides. Toutefois, cette approche serait très coûteuse car les caractéristiques de l'environnement sont recalculées avant chaque application du processus de vieillissement. Dans notre système, la carte d'humidité, notée \mathcal{W} , est générée à partir d'une carte définissant localement le taux d'humidité de la scène en présence d'eau. Cette carte est peinte à la main par l'utilisateur.

4.3.2 L'Open Markov-Chain

Le processus de vieillissement des feuilles est représenté par l'Open Markov-Chain. Dans notre système, nous l'avons implémenté sous la forme d'un graphe. Les nœuds de ce graphe, notés \mathcal{S}_i , $0 \leq i < n$ où n est le nombre de nœuds, représentent l'état dans lequel peut se trouver une feuille. Chaque nœud stocke une référence vers le modèle de feuille de l'atlas correspondant à un état donné. Les nœuds du graphe contiennent également des paramètres définissant la sensibilité de la feuille à l'environnement.

Les arcs $\mathcal{A}(i, j)$ du graphe représentent les chemins par lesquels une feuille peut passer d'un état à un autre. Les arcs stockent la probabilité pour qu'une feuille dans un état \mathcal{S}_i donné, passe dans un autre état \mathcal{S}_j . Dans notre modèle, cette probabilité de transition, noté $\mathcal{P}_{ij}(\mathbf{e}, t)$, dépend des caractéristiques de l'environnement et du pas de temps de la simulation t (Figure 4.8).

Lorsqu'un nœud donne naissance à plusieurs arcs, leurs poids représentent la probabilité relative qu'une feuille a de changer d'état. De plus, chaque colonne de la matrice de transition

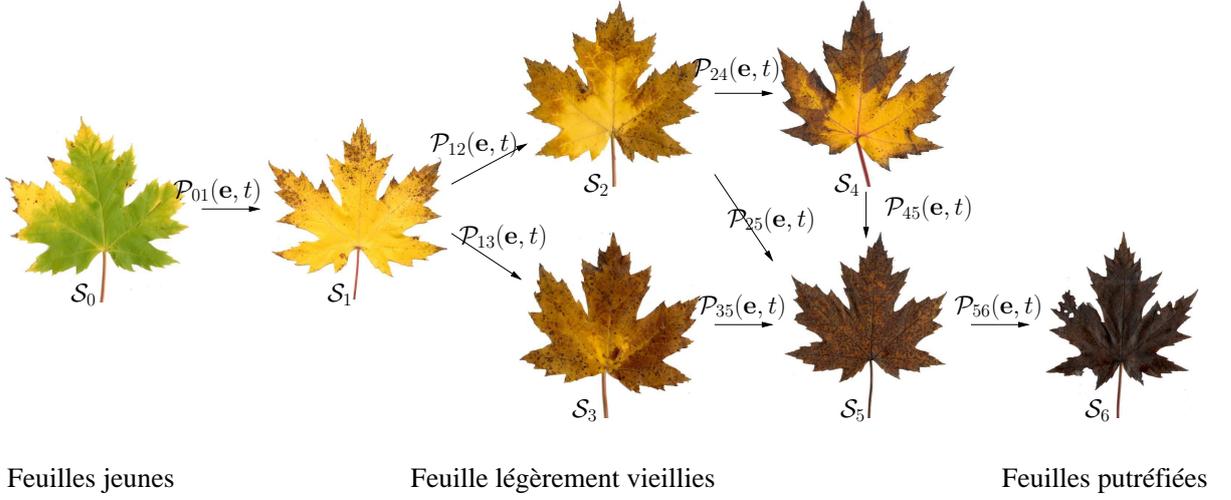


FIG. 4.8 – Représentation simplifiée de l'Open Markov Chain représentant les différents états pouvant être atteints lors du processus de vieillissement.

\mathcal{P} constituée par les probabilités $\mathcal{P}_{ij}(\mathbf{e}, t)$ doit avoir une somme unitaire :

$$\sum_{j=0}^{j=n-1} \mathcal{P}_{ij}(\mathbf{e}, t) = 1$$

Pour exprimer la probabilité qu'une feuille change d'état, nous nous sommes inspirés de la notion de demi-vie que l'on retrouve en radioactivité. L'évolution de la radioactivité est modélisée selon le principe de demi-vie qui définit le temps nécessaire pour que la moitié des éléments radioactifs se désintègrent.

Si l'on considère une feuille dans un état donné \mathcal{S}_i , nous définissons sa demi-vie, notée τ_i , comme le temps nécessaire pour que la moitié des feuilles qui se trouve dans cet état se dégrade en un autre état. Le temps nécessaire pour qu'une feuille se dégrade, noté λ_i , est relié à la demi-vie par la relation suivante :

$$\lambda_i = \frac{\ln 2}{\tau_i}$$

Dans notre système, la demi-vie d'une feuille dépend des données de l'environnement. Elle varie en fonction de la température et de la sécheresse de la feuille et est définie par la fonction $\tau_i(\mathbf{e})$. La fonction de dégradation qui lui correspond est notée $\lambda_i(\mathbf{e})$. Enfin, nous définissons la probabilité pour qu'une feuille ne change pas d'état \mathcal{S}_i après une période de t unité de temps par :

$$\mathcal{P}_{ii}(\mathbf{e}, t) = e^{-\lambda_i(\mathbf{e}) t} \quad 0 \leq i < n$$

Notre modèle permet de définir pour chaque état de la feuille une fonction de dégradation différente. Ainsi, statistiquement, les feuilles qui sont dans des états différents vieilliront à une vitesse différente. Pour chaque arc $\mathcal{A}(i, j)$ du graphe codant notre modèle de l'Open Markov Chain, les probabilités $\mathcal{P}_{ij}(\mathbf{e}, t)$ sont définies par une fonction de transition notée $X_{ij}(\mathbf{e})$. Cette

fonction définie la distribution relative d'une feuille dans un état donné \mathcal{S}_i vers un autre état \mathcal{S}_j du graphe. La probabilité pour qu'une feuille vieillisse sur une période t est donnée par $1 - \mathcal{P}_{ii}(\mathbf{e}, t)$. Ainsi, $\mathcal{P}_{ij}(\mathbf{e}, t)$ est définie par :

$$\mathcal{P}_{ij}(\mathbf{e}, t) = (1 - \mathcal{P}_{ii}(\mathbf{e}, t)) X_{ij}(\mathbf{e}) \quad 0 \leq i < n \quad i \neq j$$

Afin que la fonction $\mathcal{P}_{ij}(\mathbf{e}, t)$ puisse être définie comme une loi de probabilité, les fonctions de transitions $X_{ij}(\mathbf{e})$ doivent vérifier la relation suivante :

$$\sum_{j=0}^{j=n-1} X_{ij}(\mathbf{e}) = 1$$

Influence de l'environnement Les fonctions de demi-vie $\tau_i(\mathbf{e})$ sont définies par une interpolation bilinéaire de quatre coefficients. Ces coefficients notés $\tau_i(0,0)$, $\tau_i(1,0)$, $\tau_i(0,1)$ et $\tau_i(1,1)$ représentent, respectivement, la demi-vie d'une feuille dans chacun des cas extrêmes suivants : humide et froid, humide et chaud, sec et froid et enfin, sec et chaud. Si l'on note respectivement a et b le taux de chaleur et le taux d'humidité de la zone où se trouve la feuille, alors sa demi-vie est donnée par l'équation :

$$\tau_i(a, b) = (1 - a)(1 - b) \tau_i(0,0) + a(1 - b) \tau_i(1,0) + (1 - a)b \tau_i(0,1) + ab \tau_i(1,1)$$

De même, les fonctions de transition $X_{ij}(\mathbf{e})$ sont définies par une interpolation bilinéaire de quatre constantes, notées d_1 , d_2 , d_3 et d_4 , qui définissent la répartition relative du passage d'un état à l'autre dans chacun des cas extrêmes précédents. La fonction $X_{ij}(a, b)$ associée à des taux de chaleur a et d'humidité b est définie par :

$$X_{ij}(a, b) = (1 - a)(1 - b) d_1 + a(1 - b) d_2 + (1 - a)b d_3 + ab d_4$$

Cela implique que le designer doit définir quatre constantes pour chaque arc $\mathcal{A}(i, j)$ et quatre autres pour chaque état \mathcal{S}_i de l'Open Markov Chain.

Processus de vieillissement des feuilles Etant donné une période de temps t et une feuille dans un état initial \mathcal{S}_i , nous évaluons la probabilité de transition $\mathcal{P}_{ij}(\mathbf{e}, t)$ en tenant compte de la température locale et du taux d'humidité. Les caractéristiques de l'environnement sont données par les cartes de température et d'humidité. Leur valeur est évaluée au centre de la feuille. Ensuite, nous testons le changement d'état de la feuille en accord avec sa probabilité de transition. Si la feuille change d'état, nous mettons à jour les références vers l'instance de feuille correspondant dans l'Atlas. Dans le cas contraire, la feuille n'évolue pas.

4.4 Simulation de la chute de feuilles

Nous disséminons les feuilles dans la scène en quatre étapes (Figure 4.9). Tout d'abord, nous évaluons pour chaque feuille sa probabilité de se détacher de sa branche en fonction de

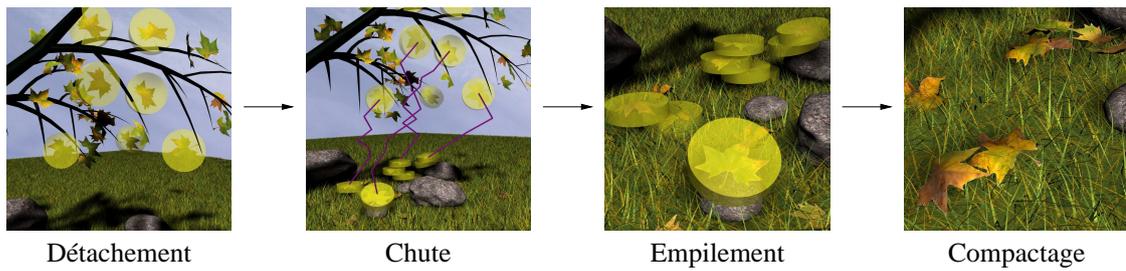


FIG. 4.9 – Ensemble des processus mis en jeu lors de la simulation de la chute des feuilles.

la vitesse du vent mais aussi de son état actuel. Ensuite, nous calculons le mouvement des feuilles qui planent dans le vent, et qui roulent et glissent sur le sol. Pour cela, nous appliquons des modèles de trajectoires procédurales. Ensuite, nous contrôlons la fin de leur déplacement par un processus d'empilement. Finalement, nous compactons les tas de feuilles au sol par le biais d'une simulation physique.

4.4.1 Processus de détachement

Dans la nature, une feuille se détache de sa branche porteuse du fait de la formation d'un joint entre la tige et la branche. La photosynthèse ne peut donc plus se faire car la feuille ne dispose plus des éléments nécessaires ; elle finit par mourir. De plus, toutes les feuilles n'ont pas la même résistance au vent. Cela dépend de leur exposition d'une part mais aussi de leur âge. Plus la feuille vieillie et plus elle va être arrachée facilement.

Nous définissons, pour chaque état dans lequel peut se trouver une feuille dans l'Open Markov Chain, une fonction qui définit sa probabilité de tomber. Cette fonction est notée $\mathcal{D}_i(s, t)$, où s définit la force du vent.

Nous définissons la fonction $\mathcal{D}_i(s, t)$ par l'équation suivante :

$$\mathcal{D}_i(s, t) = e^{-\delta_i t} e^{-s^2/\sigma_i^2}$$

La constante δ_i est définie par $\delta_i = \ln 2/\tau_i$, où τ_i représente la période de temps nécessaire pour que la moitié des feuilles tombe lorsqu'il n'y a pas de vent. La constante σ_i contrôle la sensibilité à la force du vent.

Le processus de détachement se fait comme suit : pour chaque feuille de la scène, nous calculons sa probabilité de se détacher $\mathcal{D}_i(s, t)$ en fonction de son âge et de la vitesse du vent. Si la feuille se détache, elle est ajoutée à la pile de feuilles volantes et fera l'objet des processus de chute et de dépôt. Dans le cas contraire, elle reste attachée à l'arbre.

4.4.2 Simulation de la chute des feuilles

Les mouvements que peuvent réaliser les feuilles sont définis par un ensemble de modèles de trajectoires qui peuvent être appliquées en fonction de la taille des feuilles et de la vitesse du

vent. Chaque modèle de feuille est caractérisé par un ensemble de motifs de mouvement. Cette méthode nous permet de produire des mouvements différents en fonction du type de feuille. Ainsi, une feuille de platane ne se mouvra pas de la même manière qu'une feuille de chêne. Dans notre système, nous distinguons les mouvements selon qu'ils s'agissent de mouvements de chute, ou bien des mouvements au sol.

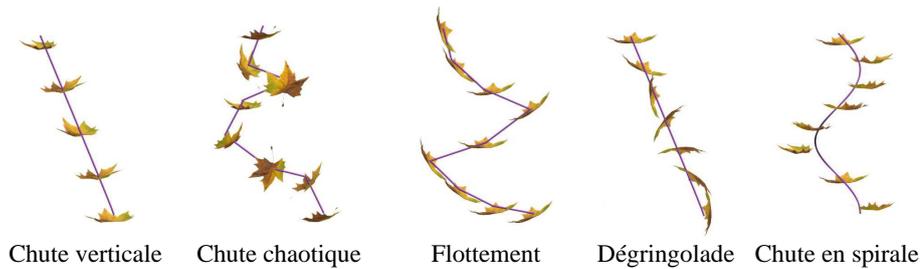


FIG. 4.10 – Motifs de chutes prédéfinis.

Trajectoires de chute Nous avons reproduit dans notre système les motifs de chutes que l'on retrouve dans certaines simulations [85, 10] et que l'on peut observer dans la nature : chute verticale, chute chaotique, flottement, dégringolade et chute en spirale (Figure 4.10). Ces trajectoires de chutes sont codées dans notre système comme un ensemble de vecteurs de déplacement relatifs. Ces vecteurs représentent la vitesse et la direction des feuilles pour chaque pas de temps. Le mouvement complet suivi par la feuille est obtenu en sommant ces vecteurs de déplacements avec les vecteurs de vitesse, qui définissent le flux d'air, pour chaque pas de temps.

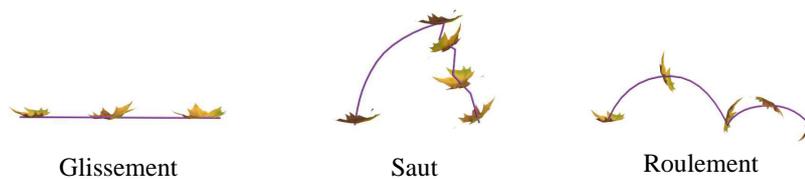


FIG. 4.11 – Motifs de déplacement des feuilles qui sont au sol.

Mouvements au sol Lorsque les feuilles sont au sol, elles glissent, décollent et roulent. Ces mouvements sont très complexes à reproduire par une simulation physique à cause des nombreuses collisions qu'ils génèrent, mais aussi à cause des flux venteux qui sont plus turbulents à ce niveau. Sous un vent léger, les feuilles peuvent glisser, ou rouler, en fonction de la nature du sol. Lorsque le vent est plus important, les feuilles peuvent être arrachées du sol et emportées dans un tourbillon. Quelques instants après, elles vont se redéposer un peu plus loin sur le sol (Figure 4.11). Nous modélisons ce mouvement en adaptant le modèle présenté dans [68] qui

simule le phénomène de saltation de grains de sable et qui consiste en une projection de grains de sable par le vent, suivi par un phénomène de dépôt.

Le principe est le suivant. Dans un premier temps, les feuilles au sol, situées sur le dessus de la pile, sont marquées comme volantes. Si le vent est léger, nous leur appliquons un mouvement de glissement dans la direction du souffle. Afin d'éviter des problèmes de collisions avec les feuilles situées dessous, nous appliquons d'abord un léger mouvement vertical. Si le vent est plus fort, nous déplaçons les feuilles dans la direction du vent et d'une hauteur proportionnelle à la force du souffle. Puis, nous appliquons un mouvement chaotique pour faire chuter la feuille sur le sol.

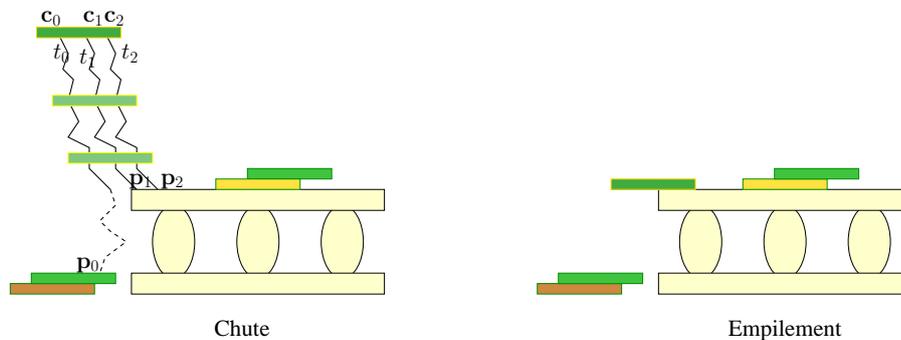


FIG. 4.12 – Processus d'empilement de feuilles.

Processus d'empilement Les feuilles ont une géométrie complexe, ce qui rend la détection de collision avec les autres feuilles, mais aussi avec les objets de la scène, coûteux en termes de temps de calcul. Pour régler ce problème, nous utilisons des cylindres englobants pour accélérer le processus de détection de collisions lors de la chute des feuilles. Chaque cylindre est échantillonné par n points d'échantillonnage, noté c_i . Ces points sont prédéfinis en fonction du type et de la géométrie de la feuille contenue dans le cylindre. La trajectoire de la feuille est discrétisée en un ensemble de segments. Nous approximations la détection de collisions en calculant l'intersection entre les trajectoires, notées t_i , de ces échantillons et les objets de la scène (Figure 4.12). Si l'on considère un segment constituant un élément de trajectoire, la détection de collision, entre une feuille et la scène, consiste à calculer l'intersection entre chacune des trajectoires suivies par les points c_i et les triangles de la scène. De façon similaire, nous détectons la collision entre deux feuilles en calculant l'intersection entre chacune des trajectoires des points c_i et les cylindres englobants des feuilles au sol. Lorsque des collisions sont détectées le long des trajectoires t_i , la feuille est figée au niveau du point d'intersection p_k si t_k est la trajectoire vérifiant l'inégalité $p_k > p_i$ ou $i \in [0, n]$.

4.4.3 Compactage des tas de feuilles

A la fin du processus de chute, nous obtenons un tas dans lequel chaque feuille plane légèrement par rapport aux autres. L'utilisation des cylindres englobants pour accélérer la détection de collisions ne permet pas de produire des amoncellements compacts. Cette dernière étape a pour objectif de générer des tas de feuilles plus réalistes. Pour cela, nous réalisons une simulation physique de la chute des feuilles à partir de cet état quasi-statique.

La complexité de la forme des feuilles et le grand nombre de feuilles de la scène rend cette étape coûteuse en termes de temps de calcul. Plusieurs techniques ont été proposées pour manipuler un grand nombre d'objets déformables en contact [46].

Toutefois, notre approche permet d'accélérer considérablement cette étape. A partir de l'état quasi-statique obtenu par l'algorithme précédent, nous réalisons la simulation de chute de manière incrémentale. Pour cela, nous partitionnons les tas en ensembles de feuilles organisés en fonction de leur hauteur respective. La scène finale est obtenue en simulant la chute de chaque ensemble de feuilles, étage par étage, et en les figeant après la simulation.

4.5 Résultat et discussion

L'Open Markov Chain permet de représenter le cycle de vieillissement des feuilles en tenant compte des propriétés de l'environnement. Notre système calcule automatiquement la distribution de feuilles sur le sol lorsqu'elles tombent de l'arbre et génèrent des tas. Nous utilisons un atlas pour stocker les modèles de feuilles à différents instants de leur vieillissement. Dans cette section, nous présentons les résultats que nous avons obtenus en modélisant des scènes d'automne avec l'Open Markov Chain.

Les figures 4.13, 4.14 et 4.15 montrent les résultats que nous avons obtenus en modélisant des paysages d'automne avec notre système.

La scène de la figure 4.13 contient 9142 feuilles réparties dans les deux arbres. A chaque pas de simulation, l'algorithme de sélection détache des feuilles qui vont alors chuter jusqu'au sol. Du fait de la présence de vent, les feuilles s'accumulent contre le mur et forment plusieurs tas. On peut constater que la distribution de la coloration des feuilles est réaliste. Pour cette scène, nous n'avons utilisé que 16 modèles de feuilles et défini 32 transitions possibles. Cela montre que l'on peut se contenter d'un très petit nombre de modèles pour produire des scènes dont l'aspect est plausible d'un point de vue biologique.

Toutefois, dans la nature, lorsque des feuilles forment un tas aussi dense, elles accumulent beaucoup d'humidité qui accélère le phénomène de putréfaction. De plus, cette humidité va assouplir les feuilles qui vont alors épouser le relief du sol. Dans notre système, les feuilles ne modifient pas les données de l'environnement. De ce fait, l'accumulation des feuilles n'accélère pas le processus de putréfaction. De plus, comme nous utilisons des instances de feuilles, nous ne pouvons pas modifier leur géométrie selon des critères locaux. Dans ce cas, on produirait une très grande quantité de feuilles qui serait difficilement gérable. Toutefois, cela pourrait



FIG. 4.13 – Scène virtuelle illustrant l'évolution de la coloration et de la chute des feuilles au cours de la simulation et en accord avec les conditions climatiques.

être réalisé par le biais d'une simulation physique en considérant les feuilles comme des corps mous.

Age	n_F	t	n_I	n_S
1	0	1'40	0	0
2	218	2'	3127952	1384
3	647	4'95	6406260	2850
4	1156	10'90	10291244	4400
5	1735	20'25	14129548	7598
6	2381	34'06	18888808	10700
7	3066	47'95	23213457	14058
8	3725	59'96	27260268	17685
9	4340	74'39	32132686	21680
10	4957	85'20	36198222	25697
11	5482	96'26	40619604	29991
12	5947	108'98	44916299	34490
13	6424	118'86	49065220	39059
14	6808	126'86	53088893	43664
15	7136	138'15	57095990	48382
16	7439	140'76	61265993	53202

TAB. 4.1 – Statistiques obtenues à partir de la simulation de la scène illustrée par la figure 4.13.

La table 4.1 donne des statistiques liées à cette scène : le temps écoulé lors de la simulation, le nombre de feuilles qui tombent n_F , le temps de simulation du vieillissement et de la chute des cylindres englobants t , le nombre d'intersections testées n_I et le nombre de segments constituant les trajectoires n_S générées par les feuilles lors de leur chute. Cette table montre que la simulation du vieillissement des feuilles et leur positionnement à partir des cylindres englobants est rapide. La scène finale, obtenue après la simulation de l'effondrement des tas peut nécessiter jusqu'à 12h de calcul. Ces temps montrent que l'étape durant laquelle les feuilles sont compactées est la plus coûteuse en temps de calcul. Toutefois, cela n'est pas un problème car il s'agit de l'étape finale de l'algorithme. De plus, l'utilisateur bénéficie d'un retour rapide du pré-positionnement des feuilles par le biais de l'algorithme de chute qui offre une approximation de la scène finale. Il peut ainsi modifier les paramètres de vieillissement du modèle ou les conditions environnementales, tout en ayant un retour visuel rapide du résultat final.

Les scènes de la figure 4.14 et 4.15 contiennent entre 14302 et 25202 feuilles. La simulation du vieillissement des feuilles a été réalisée en moins d'une seconde. L'étape de préplacement a pris quant à elle moins de 3 minutes. Pour modéliser ces deux scènes, nous avons utilisé des outils de contrôle pour ajouter des feuilles dans des zones précises : la brouette et la fontaine. L'algorithme automatique de chute de feuilles à partir des données de l'environnement est bien



FIG. 4.14 – Feuilles tombant d'un platane, recouvrant le sol et s'accumulant dans une brouette.



FIG. 4.15 – Simulation de la chute de feuilles de bouleau dans une scène virtuelle.

adapté pour distribuer rapidement une grande quantité de feuilles mais n'est pas suffisant. En effet, dans le cadre d'un travail artistique, l'utilisateur peut souhaiter ajouter ou enlever des feuilles dans certaines zones de la scène indifféremment de toute considération environnementale. Par exemple, cela permet de chasser les feuilles d'une zone de passage telle qu'une route ou un trottoir. Nous avons commencé à intégrer trois outils dans notre système : un outil de génération de feuille pour disperser des feuilles à un endroit précis de la scène, un outil de génération de souffle et d'explosion pour dégager des zones et, enfin, un outil de création de vortex pour créer des accumulations de feuilles.

Afin de permettre la gestion en mémoire d'une très grande quantité de feuilles, nous utilisons un atlas de modèles de feuilles. La scène est alors constituée d'un ensemble d'instances définies par une simple matrice de transformation 4×4 et d'une référence à un modèle de l'atlas. Si cette technique présente l'avantage de minimiser le coût de la scène en mémoire, elle a un inconvénient majeur.

Comme la géométrie des feuilles est définie par les modèles de l'atlas, lors du processus de vieillissement, certaines feuilles peuvent changer significativement de forme. A chaque étape du vieillissement, nous devrions donc recalculer toute la simulation de chute afin qu'aucune feuille ne s'intersecte. Nous avons développé l'algorithme de chute pour éviter cela. Celui-ci génère un empilement de cylindres englobants de sorte que lorsque ceux-ci sont en contact, les feuilles ne s'intersectent pas. Pour cela, le cylindre englobant doit englober toutes les feuilles de l'atlas. De fait, il peut arriver que le cylindre soit si haut qu'un tas de feuilles atteigne une feuille attachée à un arbre. Dans ce dernier cas, nous commençons par réaliser la simulation physique sur les feuilles des tas puis nous appliquons notre algorithme de chute aux feuilles restées dans l'arbre en considérant le nouveau tas de feuilles comme faisant partie de la scène statique. Enfin, la scène finale est obtenue en effondrant les nouveaux tas obtenus par le biais d'une simulation physique. Le problème de cette approche est qu'elle augmente encore le temps nécessaire à l'étape d'effondrement.

Chapitre 5

Création de fissures réalistes

Sommaire

5.1	Modélisation des fissures	76
5.1.1	Modèle de fissures génériques	76
5.1.2	Edition d'un motif de fissure	77
5.2	Ajout d'une fissure	78
5.2.1	Création du squelette	79
5.2.2	Création du volume de la fissure	80
5.2.3	Cas particuliers	80
5.2.4	Résultats	83
5.3	Animation de la propagation d'une fissure	88
5.4	Conclusion	89

Publication Ces travaux ont fait l'objet d'une publication à la conférence SMI (Shape Modeling International) en 2004, intitulé *Procedural Modeling of cracks and fractures* [4]; Et d'une publication à la conférence Pacifique Graphics, en 2005, sous le titre *Modeling cracks and fractures* [2].

Dans la nature, les fissures sont présentes sur un très grand nombre d'objets. Leur apparition peut être due à des chocs ou des impacts mais aussi à des phénomènes thermiques telle qu'une succession de dilatations et de contractions des matériaux. De ce fait, en informatique graphique, la création de fissures a principalement été traité comme un phénomène physique. En tant que tel, les méthodes proposées sont toujours orientées vers la simulation. Toutefois, cette approche pose un grand nombre de problèmes.

Tout d'abord, la simulation d'un phénomène physique nécessite très souvent un très grand nombre de paramètres, généralement issus de mesures. Ces paramètres sont souvent techniques et à ce titre, ils sont difficiles à manipuler dès lors que l'on recherche un résultat précis. De plus, dans certains cas, une simple mesure ne permet pas de rendre totalement compte de la richesse d'un objet. Par exemple, lorsque celui-ci est constitué de matériaux granuleux, dont la résistance et la répartition variable de chaque type de grain à une grande importance dans la trajectoire suivie par une fracture. Pour régler ce genre de problème, il est nécessaire de passer par la modélisation du matériau, ce qui reste une étape très complexe.

Ensuite, la simulation de l'apparition d'une fissure sur un objet de synthèse demande des temps de calculs conséquents. Ce temps est justifié dans le cas de l'évaluation de la résistance d'un édifice architectural à la rupture. Ce n'est pas le cas en ce qui concerne la création ou l'animation de l'apparition d'une fissure dans le cadre d'une œuvre artistique, comme un film d'animation. L'artiste a besoin d'un retour visuel direct pour contrôler son travail et doit pouvoir obtenir le résultat qu'il souhaite avec le plus de confort possible.

Enfin, une caractéristique importante de la simulation de fissures est qu'elle nécessite une discrétisation des objets. Dans la grande majorité des travaux existants, cette discrétisation a deux conséquences. La première est qu'elle produit des artefacts le long du front de propagation de la fissure sous la forme de crénelage. Celui-ci est d'autant plus marqué que les tensions locales sont élevées. Pour diminuer ces défauts, on peut augmenter la finesse de la discrétisation, soit localement, soit sur tout l'objet. Ceci pose alors un autre problème car on augmente ainsi le coût de la simulation, d'une part en mémoire et d'autre part en termes de temps de calcul.

Notre objectif est de modéliser des fissures réalistes même vue de très près. Pour cela, nous souhaitons pouvoir modéliser une grande variété de fissures : fines, larges, lissées ou encore extrêmement bruitées. Il est également nécessaire de pouvoir modéliser leur intérieur qui peut être visible selon certains points de vue. Les méthodes par simulation physique ne sont donc pas adaptées à notre problème. Dans ce chapitre, nous décrivons notre méthode d'ajout de fissures qui repose sur une approche phénoménologique. Celle-ci, plus proche de ce que peut faire un artiste, nous permet de résoudre un grand nombre des problèmes apportés par la simulation. Notre modèle permet la création de la vraie géométrie des fissures, y compris l'intérieur. Ainsi, il n'est pas nécessaire de discrétiser les objets ce qui permet d'éviter l'apparition d'artefacts le long de la fissure. Notre modèle fonctionne pour un très grand nombre de matériaux et n'est pas limité aux objets constitués de matériaux homogènes.

La figure 5.1 illustre notre approche. Dans notre système, une fissure est définie par un

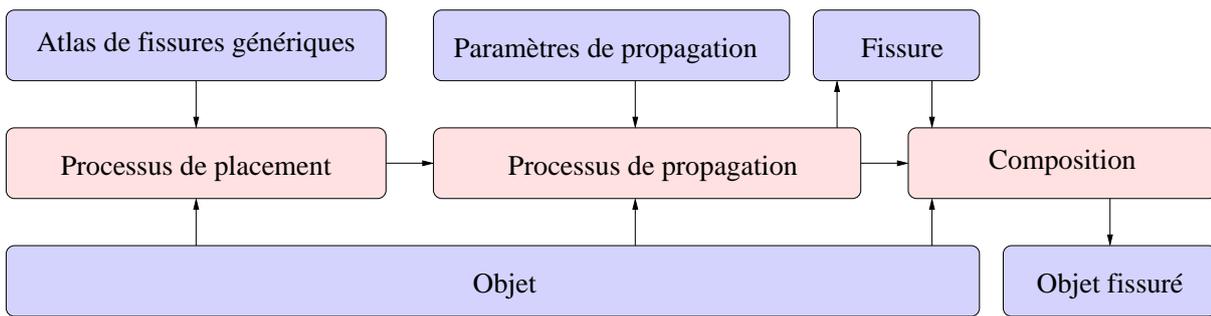


FIG. 5.1 – Aperçu du processus d’ajout d’une fissure sur un objet.

modèle générique défini en deux dimensions (section 5.1) stocké dans un atlas. Celui-ci est ensuite projeté sur un objet pour produire le volume de la fissure (section 5.2). L’objet fissuré est obtenu en retranchant ce dernier au volume de l’objet initial.

5.1 Modélisation des fissures

Les fissures sont représentées par des fissures génériques en deux dimensions qui permettent de décrire leur structure et leur forme intérieure. Nous stockons ces fissures génériques dans un atlas organisé en fonction du matériau auquel elles peuvent être appliquées. Cette méthode nous permet de générer aisément une grande variété de fissures définies dans l’atlas par exemple par le biais d’opération de bruitage ou de lissage.

5.1.1 Modèle de fissures génériques

Une fissure générique $\mathcal{F} = \{\mathcal{M}, \mathcal{P}\}$ est caractérisée par un motif de fissure \mathcal{M} et un ensemble de courbes de profils \mathcal{P} qui définissent respectivement sa structure et sa forme intérieure au niveau de ses sections.

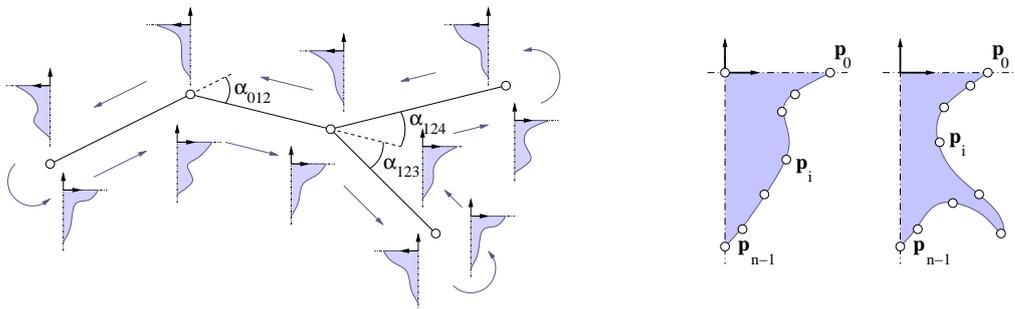


FIG. 5.2 – Structure d’une fissure générique. Les flèches illustrent le processus de parcours du graphe.

Les motifs de fissure sont représentés sous la forme d’un graphe orienté dénoté \mathcal{G} . Les nœuds du graphe notés \mathcal{N} , stockent les profils qui définissent l’intérieur des différentes sec-

tions de la fissure. Chaque nœud du graphe stocke, deux ou plusieurs profils, en fonction de sa connectivité. Les feuilles du graphe qui sont des nœuds unaires et les nœuds binaires stockent deux profils. Les nœuds n-aires tels que $n > 2$ possèdent n profils (Figure 5.2).

Notons \mathcal{N}_j une feuille et \mathcal{N}_i son nœud père. \mathcal{N}_j stocke deux profils \mathcal{P}_{ij} et \mathcal{P}_{ji} . Dans le cas général, \mathcal{P}_{ijk} fait référence au profil du nœud \mathcal{N}_j lorsque nous traversons le graphe de \mathcal{N}_i à \mathcal{N}_k .

L'arc entre \mathcal{N}_i et \mathcal{N}_j sera noté \mathcal{A}_{ij} . L'angle formé par deux arcs consécutifs \mathcal{A}_{ij} et \mathcal{A}_{jk} sera noté α_{ijk} . L'arc \mathcal{A}_{ij} est valué par la distance entre deux nœuds \mathcal{N}_i et \mathcal{N}_j notée δ_{ij} . La distance entre deux nœuds du graphe est définie par la distance minimum nécessaire pour les joindre en parcourant les arcs.

Nous définissons un profil de fissure par une spline. Les points de contrôle seront notés \mathbf{p}_i , $i \in [0, n - 1]$, n étant le nombre de points de contrôle.

5.1.2 Edition d'un motif de fissure

Il y a deux façons pour définir le motif d'une fissure : à partir d'une photographie ou procéduralement. Dans la pratique, nous combinons ces deux méthodes.

Nous avons développé une interface (figure 5.3) qui permet au designer d'ajouter et de retirer interactivement des points de contrôles à partir d'une photographie de fissure. Ces points correspondent aux nœuds du motif de fissure. L'angle relatif α_{ijk} entre les arcs \mathcal{A}_{ij} et \mathcal{A}_{jk} , et la distance δ_{ij} , sont calculés directement à partir de ce graphe en deux dimensions. Ensuite, l'utilisateur peut définir pour quelques nœuds un ou plusieurs profils en fonction de sa valence. Le designer doit définir au minimum un profil pour chaque nœud terminal. Les autres profils seront générés par interpolation.

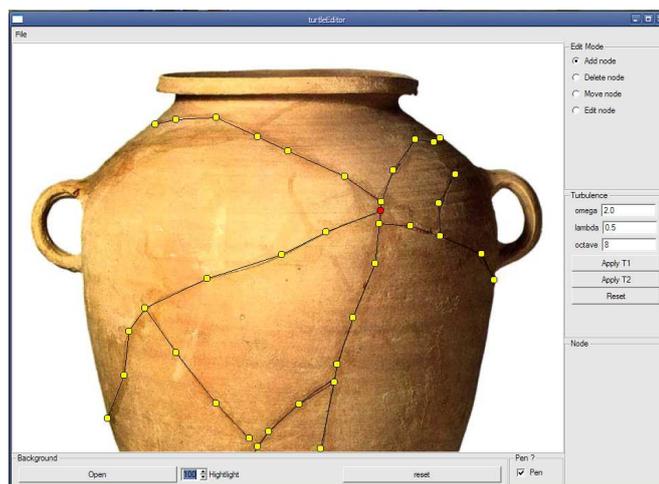


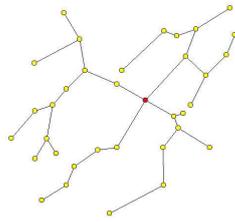
FIG. 5.3 – Création d'un motif à partir de la photographie d'une fissure.

Les motifs de fissures peuvent aussi être générés procéduralement. A partir d'un motif grossier édité à la main ou par le biais d'un L-System, l'utilisateur peut générer des motifs détaillés. Nous avons développé des outils de hauts niveaux qui permettent de perturber (figure 5.4) et,

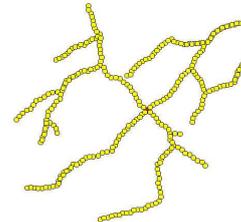
ou, de lisser (figure 5.5) un motif de fissure.



Fissure de pierre



Fissure générique

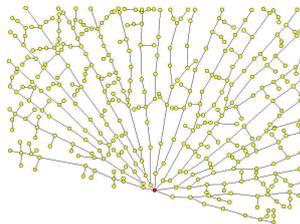


Motif bruité

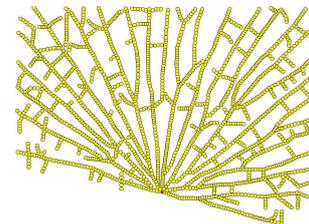
FIG. 5.4 – Création d'un motif de fissure de type pierre par bruitage.



Fissure de verre



Fissure générique



Motif lissé

FIG. 5.5 – Création d'un motif de fissure de type verre par lissage.

Nous utilisons deux méthodes pour perturber le motif. La première méthode consiste à insérer des points le long des arcs \mathcal{A}_{ij} du motif et à perturber leur position le long de celui-ci. Dans ce cas, le motif grossier *porte* le motif perturbé. La deuxième méthode que nous utilisons, consiste à insérer des points le long des arcs et à perturber tout le graphe par une fonction de bruit. Dans ce cas, tout le motif est perturbé. Ces méthodes permettent à l'utilisateur de créer des fissures d'aspect rugueux à partir d'un motif de fissure grossier. Nous utilisons cette approche pour créer des fissures sur des objets en terre ou en pierre.

Pour lisser le motif, nous utilisons une fonction de perturbation pour contrôler la variation entre les angles α_{ijk} et α_{jkl} intervenant au niveau des nœuds \mathcal{N}_{ij} , \mathcal{N}_{jk} et \mathcal{N}_{kl} . Lorsque la variation est faible, le motif est très lisse, a contrario, lorsque les variations sont importantes, le motif s'enroule sur lui même. L'utilisation d'une fonction de perturbation permet de produire des motifs lisses qui présentent tout de même des décrochements. Nous avons utilisé cette méthode pour créer les fissures sur les objets en verre.

5.2 Ajout d'une fissure

Le processus d'ajout d'une fissure \mathcal{F} sur un objet initial \mathcal{O} peut-être décomposé en trois phases (figure 5.6). La première phase consiste à projeter le motif de fissure \mathcal{M} sur l'objet ce qui a pour but de créer un squelette en trois dimensions. Ensuite, nous créons les profils attachés aux nœuds du squelette en interpolant les profils stockés au niveau des sections du motif de

fissure, et nous les orientons en fonction de la normale locale. Le volume final de la fissure

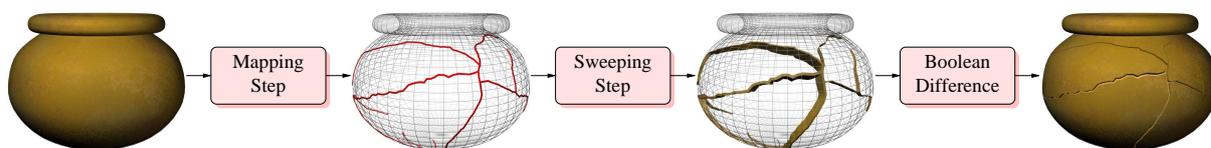


FIG. 5.6 – Processus de modélisation d'une fissure.

est obtenu à partir d'un cylindre généralisé construit en balayant les profils interpolés le long du squelette. Enfin, la dernière étape consiste à générer l'objet fissuré. Pour cela, nous ôtons le volume de la fissure à celui de l'objet original en calculant la différence booléenne $\mathcal{O} - \mathcal{V}$. Dans notre système, nous utilisons des techniques classiques pour calculer la différence booléenne.

5.2.1 Création du squelette

Etant donné une fissure générique \mathcal{F} , la création du squelette en trois dimensions lui correspondant est réalisée en projetant le motif de fissure \mathcal{M} sur la surface de l'objet.

A partir d'un point d'ancrage \mathbf{c}_0 et d'une direction initiale \mathbf{u}_0 , l'algorithme reproduit sur la surface maillée un déplacement correspondant au parcours du graphe \mathcal{G} . Les directions et les pas de déplacement sont définis à partir des paramètres α_{ij} et δ_{ij} stockés, respectivement, au niveau des nœuds et des arcs du graphe.

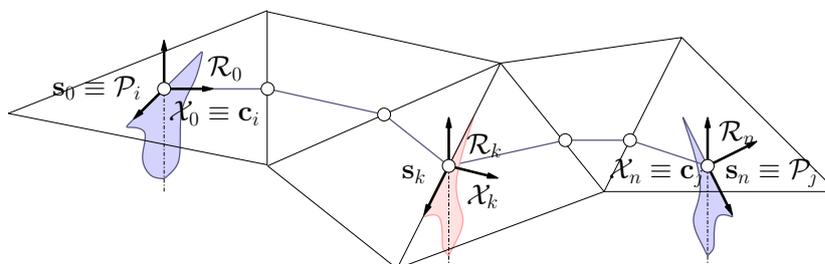


FIG. 5.7 – Structure du squelette en trois dimensions produit par la projection d'un motif de fissure.

Les nœuds \mathcal{N}_i du motif sont transformés en sommets notés \mathbf{c}_i et les arcs \mathcal{A}_{ij} sont transformés en un ensemble de segments $[\mathbf{s}_k, \mathbf{s}_{k+1}]$, $k \in [0, n - 1]$ avec $\mathbf{s}_0 \equiv \mathbf{c}_i$ et $\mathbf{s}_n \equiv \mathbf{c}_j$. Les sommets \mathbf{s}_k correspondent à l'intersection entre les arêtes des triangles du maillage et le motif de la fissure qui se propage sur l'objet (figure 5.7).

Pour chaque sommet \mathbf{s}_k du squelette, nous construisons un repère local $\mathcal{R}_k = (\mathbf{s}_k, \mathbf{d}_k, \mathbf{t}_k, \mathbf{n}_k)$. Le vecteur \mathbf{d}_k définit la direction de propagation de la fissure au niveau du sommet \mathbf{s}_k et \mathbf{n}_k est la normale à la surface en ce point. Si \mathbf{s}_k est à l'intérieur d'un triangle alors, \mathbf{n}_k est la normale de celui-ci. A contrario, si \mathbf{s}_k est sur une arête ou est confondu avec un sommet, alors, \mathbf{n}_k est

définie comme la moyenne des normales des triangles voisins. Le vecteur tangent \mathbf{t}_k est défini par $\mathbf{t}_k = \mathbf{n}_k \wedge \mathbf{u}_k$.

5.2.2 Création du volume de la fissure

Les sections \mathcal{X}_k du volume de la fissure, situées au niveau des sommets \mathbf{s}_k , sont obtenues en interpolant les profils définis en $\mathbf{s}_0 \equiv \mathbf{c}_i$ et $\mathbf{s}_n \equiv \mathbf{c}_j$, et en les replaçant dans le repère \mathcal{R}_k . Pour réaliser l'interpolation entre deux profils, nous évaluons la distance δ_{ik} entre deux sommets \mathbf{c}_i et \mathbf{c}_j du motif. Nous définissons cette distance comme la somme des distances de chaque segment entre $\mathbf{s}_0 = \mathbf{c}_i$ et \mathbf{s}_k :

$$\delta_{ik} = \sum_{j=0}^{k-1} \|\mathbf{s}_{j+1} - \mathbf{s}_j\|$$

Ainsi, pour chaque sommet \mathbf{s}_k , nous définissons sa section par l'interpolation linéaire suivante :

$$\mathcal{X}_k = \frac{\delta_{ik}}{\delta_{ij}} \mathcal{P}_i + \left(1 - \frac{\delta_{ik}}{\delta_{ij}}\right) \mathcal{P}_j$$

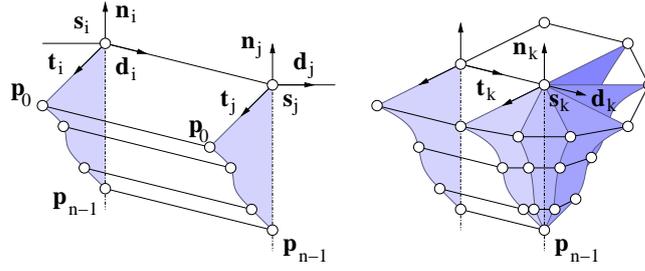


FIG. 5.8 – Création du volume de la fissure.

Pour créer le volume de la fissure, nous parcourons le squelette comme illustré sur la figure 5.2. Pour chaque segment $[\mathbf{s}_k, \mathbf{s}_{k+1}]$ du squelette, le volume de la fissure est défini par le cylindre généralisé qui connecte les sections \mathcal{X}_k et \mathcal{X}_{k+1} (Figure 5.8). Nous générons le volume des nœuds terminaux du squelette en interpolant les deux profils autour de l'axe qui passe par le sommet \mathcal{X}_{s_k} et qui a pour direction \mathbf{n}_k (Figure 5.8).

Après avoir construit le volume de la fissure, nous générons l'objet fissuré. Pour cela, nous retirons le volume de la fissure de l'objet initial par différence booléenne. Dans notre système, nous utilisons des techniques standard pour réaliser cette opération.

5.2.3 Cas particuliers

Dans certains cas, deux sections consécutives \mathcal{X}_k et \mathcal{X}_{k+1} du squelette s'intersectent. Ceci a pour conséquence de produire un volume inconsistant pour le segment $[\mathbf{s}_k, \mathbf{s}_{k+1}]$ correspondant, durant la phase de génération de la géométrie. Ce cas apparaît principalement dans les zones où la courbure locale de la surface est trop importante par rapport à la taille de la section. C'est également le cas lorsque la fissure marque un changement de direction important et

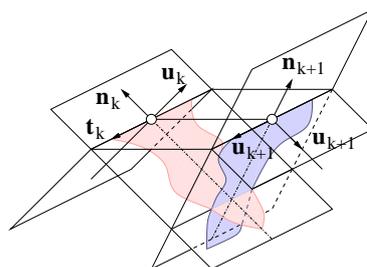


FIG. 5.9 – Une Intersection entre deux sections peut apparaître dans les zones de forte courbure.

qu'elle revient sur elle-même (Figure 5.9).

L'apparition d'un volume inconsistant nous empêche d'évaluer correctement l'opération booléenne nécessaire à la création de l'objet fissuré. Ces volumes sont dus au fait que la résolution de l'objet maillé n'est pas forcément adaptée à la résolution de la fissure. Lors de la phase de projection, chaque arc de la fissure générique est décomposé en $n + 1$ segment si $n + 1$ est le nombre d'arêtes traversées. Si la fissure est trop profonde ou trop large pour les torsions générées par la projection, alors le volume peut s'auto-interpénétrer. Pour résoudre ce problème, nous réalisons une étape de ré-échantillonnage du squelette qui consiste à fusionner et/ou décomposer les sommets du squelette de sorte que sa résolution soit adaptée à sa morphologie. Le squelette ainsi généré respecte au mieux la courbure locale et les propriétés de la fissure.

Processus de fusion Cette étape a pour rôle d'adapter localement le nombre de sommets du squelette à la taille des sections. Notons r_i et r_j les rayons des disques centrés en \mathbf{s}_i et \mathbf{s}_j , sur le squelette, et englobant les sections correspondantes \mathcal{X}_i et \mathcal{X}_j . Les deux sections risquent de s'intersecter si $\delta_{ij} < r_i + r_j$. Dans ce cas, nous fusionnons \mathbf{s}_i et \mathbf{s}_j et créons le nouveau sommet \mathbf{s}_k sur le squelette par interpolation linéaire :

$$\mathbf{s}_k = \frac{r_i \mathbf{s}_i + r_j \mathbf{s}_j}{r_i + r_j} \quad \mathcal{X}_k = \frac{r_i \mathcal{X}_i + r_j \mathcal{X}_j}{r_i + r_j}$$

Processus de décomposition Le processus de décomposition a pour objectif d'ajouter des points sur le squelette si la distance δ_{ij} est trop grande par rapport à la taille de la section. Cette étape permet de produire des squelettes avec un échantillonnage, le maillage du volume final est ainsi plus homogène. Cette propriété permet d'obtenir de meilleurs résultats lors de l'opération booléenne. Un segment du squelette défini par les sommets \mathbf{s}_i \mathbf{s}_j est décomposé s'il vérifie $r_i + r_j < 2\delta_{ij}$. Dans ce cas, nous insérons un nouveau point \mathbf{s}_k et une section \mathcal{X}_k sur le segment de la même manière que pour la fusion.

Cas des surfaces bruitées L'échantillonnage adaptatif du squelette permet de gérer des surfaces constituées d'un maillage irrégulier et présentant de fortes variations au niveau des basses (structure de la forme) et des hautes fréquences (les détails de l'objet). Toutefois, lorsque l'objet

présente beaucoup de hautes fréquences, comme une surface bruitée, l'algorithme de projection traverse beaucoup de triangles et subit de nombreuses variations de directions. Cela a deux conséquences. Tout d'abord, le squelette va présenter de fortes distorsions par rapport à la fissure générique initiale. D'autre part la phase d'échantillonnage adaptatif risque de produire un squelette beaucoup trop simplifié.

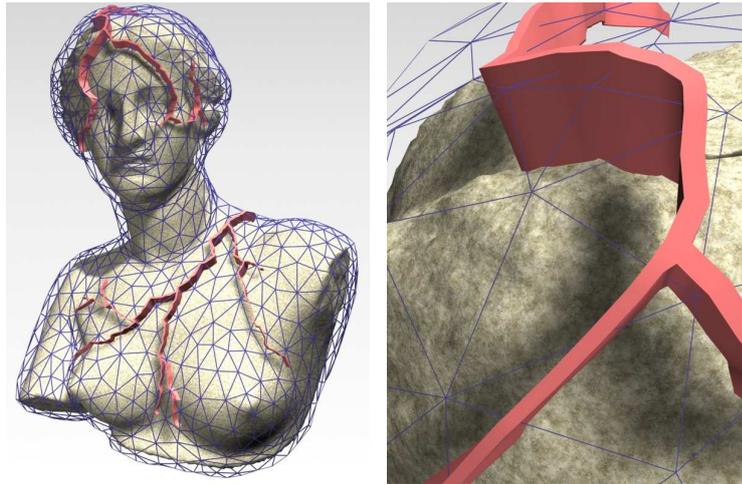


FIG. 5.10 – La fissure est construite sur le maillage lissé qui englobe l'objet initial.

Notre approche consiste à utiliser une surface lissée, notée $\tilde{\mathcal{O}}$ qui approxime et enveloppe la surface initiale (Figure 5.10). Le nouveau squelette $\tilde{\mathcal{S}}$ est obtenu en projetant le motif \mathcal{M} sur $\tilde{\mathcal{O}}$. Le volume de la fissure est généré comme décrit précédemment, et l'objet fissuré est obtenu en évaluant l'opération booléenne $\mathcal{O} - \mathcal{V}$. La surface $\tilde{\mathcal{O}}$ doit être lisse et doit envelopper l'objet initial pour éviter la formation de tunnel lors de l'opération booléenne. Nous créons cette surface à partir de l'objet initial en utilisant les techniques standard de simplification de maillage [27, 47]. Les sommets du nouvel objet situé à l'intérieur de l'objet initial sont déplacés à l'extérieur de \mathcal{O} de sorte que la surface finale de $\tilde{\mathcal{O}}$ soit intégralement englobée. La figure 5.10 illustre ce processus. La statue Aphrodite est définie par un maillage de 92188 triangles et a une surface extrêmement bruitée notamment à cause de la méthode de reconstruction utilisée. Pour ajouter des fissures sur cet objet, nous avons tout d'abord construit une version simplifiée et lissée l'enveloppant (maillage en bleu). Le volume de la fissure est généré en projetant la fissure générique sur cet objet qui est constitué de 3042 triangles seulement. Cette méthode nous permet d'ajouter des fissures même sur des objets extrêmement bruités mais également d'accélérer l'étape de projection dans le cas de maillage très dense. La seule contrainte de ce processus, est que l'objet qui sert de support doit englober l'objet à fissurer. Si cela n'est pas le cas, des tunnels peuvent apparaître lors de la création du volume de l'objet fissuré.

5.2.4 Résultats

Nous avons développé une application (figure 5.11) dans laquelle, le designer peut sélectionner un modèle de fissure dans un atlas de motifs et contrôler interactivement la localisation, l'orientation et la taille de la fissure sur l'objet. L'algorithme présenté dans la section 5.2 projette automatiquement le modèle de fissure sur l'objet et génère le volume de la fissure. L'utilisateur peut contrôler directement l'orientation de la fracture sur l'objet, éditer ses profils ou encore modifier sa forme, le tout avec un retour visuel. Nous avons utilisé cette interface pour ajouter des fissures sur une grande variété d'objets. Les images qui suivent illustrent les résultats que nous avons obtenus.

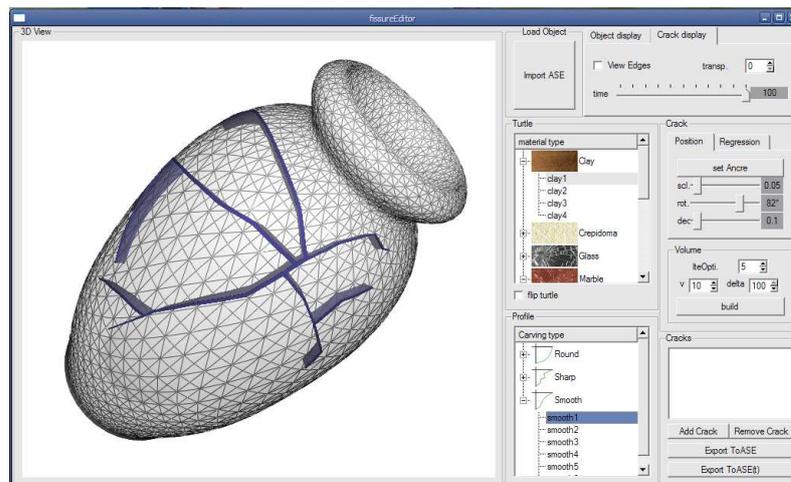


FIG. 5.11 – Interface d'édition de fissures.

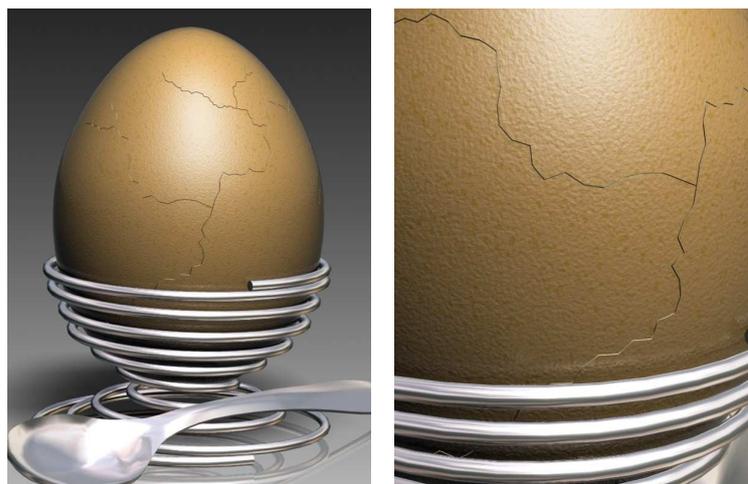


FIG. 5.12 – Coquille d'oeuf cuit à la coque fissurée.

L'ajout de la fissure sur l'œuf (Figure 5.12) a été réalisé à partir d'un motif de fissure de type pierre. Nous avons généré ce motif en partant d'un motif grossier et en lui appliquant notre

opérateur de turbulence. Le motif que nous avons obtenu reproduit l'aspect granuleux d'une coquille d'œuf. La création de cette scène à pris environ 30 minutes.



FIG. 5.13 – Vase cassé.



FIG. 5.14 – Verre de vin fissuré.

Les fissures sur le verre de vin et le vase (Figure 5.14 et 5.13) ont été réalisées en utilisant notre procédure de lissage de motifs grossiers. Cette méthode nous a permis de produire des fissures lisses de faibles épaisseurs typiques des fissures dans le verre. Il nous a fallu moins de 10 minutes pour éditer les motifs de fissures correspondants, puis les plaquer sur ces objets.

Nous avons fissuré les pieds et les chapeaux des champignons en utilisant un motif de fissure très légèrement bruité pour qu'il ne soit pas trop régulier. Ces fissures ont été réalisées en moins de 10 minutes.

Nous avons utilisé notre système pour ajouter des fissures dans une scène complexe (Figure 5.16) constituée d'un grand nombre d'objets faits de différents matériaux. La création des



FIG. 5.15 – Fissure sur des champignons.



FIG. 5.16 – Scène complexe avec différents types de fissures sur du bois, du crépis, du marbre et du verre.

motifs de fissures et leur ajout sur les différents objets de la scène ont été réalisés en moins de 1 heure 30.



FIG. 5.17 – Main courante en bois fendue dégradant le mur qui la soutient.

Nous avons fissuré le crépis (Figure 5.17) en utilisant un motif de fissure produit par notre opérateur de turbulence. Notre système d’application d’un motif de fissure nous a permis de placer très précisément cette fissure là où il est le plus probable qu’elle apparaisse : sous la rampe de l’escalier.



FIG. 5.18 – Barrière en bois abîmée par le temps.

Finalement, nous avons ajouté sur le banc, les barrières (Figure 5.18) et la rampe (Figure 5.17) des fissures de type bois. La fissure sur le haut du banc illustre le cas d’une fissure produite par un coup : la fissure est très bruitée. Les fissures sur le bas du banc, la rampe et les barrières illustrent le cas de fissures dues au temps et au travail du bois : les fissures sont lisses et se meurt progressivement. Finalement, afin d’obtenir des fissures très réalistes, nous les avons orientées dans le sens des fibres du bois.

Les fissures sur le jeu d’échec (figure 5.19) ont été générées à partir d’un motif grossier et



FIG. 5.19 – Jeu d'échec fissuré.

des opérateurs de turbulence et de lissage appliqués l'un après l'autre. Il nous a fallu moins de 5 minutes pour créer et placer la fissure.

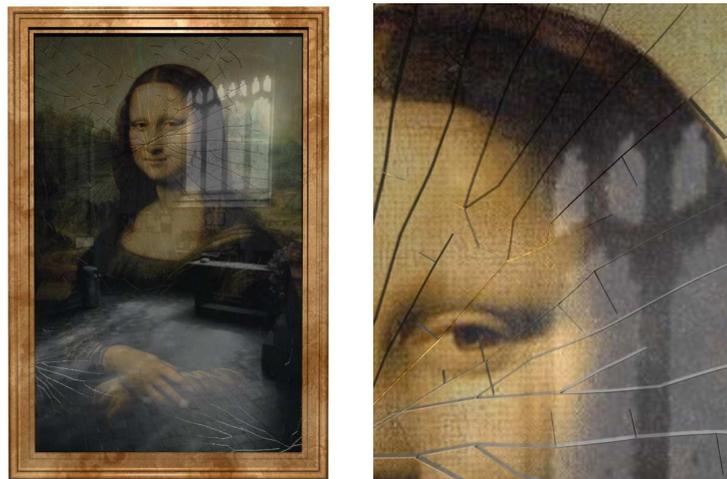


FIG. 5.20 – Peinture dans un cadre fissuré.

Les fissures sur la vitre du cadre (figure 5.20) ont été créées à partir du motif de la fissure 5.5 auquel on a appliqué l'opérateur de lissage et une opération de symétrie pour obtenir des motifs circulaires.

Sur la figure 5.21, on peut constater qu'une fissure présente sur l'œuvre d'origine a été éliminée par le processus de reconstruction alors qu'elle est toujours présente sur la texture. Nous avons utilisé notre système pour extraire la fissure de la texture et l'appliquer au modèle reconstruit.



FIG. 5.21 – Ajout d’une fissure après reconstruction.

5.3 Animation de la propagation d’une fissure

Nous simulons la propagation de la fissure en paramétrant les fissures génériques par le temps. La propagation de la fissure est obtenue en générant l’ensemble des fissures génériques correspondant à un intervalle de temps donné.

A partir d’une fissure générique statique $\mathcal{F} = \mathcal{M}, \mathcal{P}$, nous définissons un modèle générique paramétré par le temps $\mathcal{F}(t) = \mathcal{M}(t), \mathcal{P}(t)$ qui décrit comment la fissure se propage à travers \mathcal{F} . L’animation finale est obtenue en instanciant ce modèle générique à un instant donné t_0 .

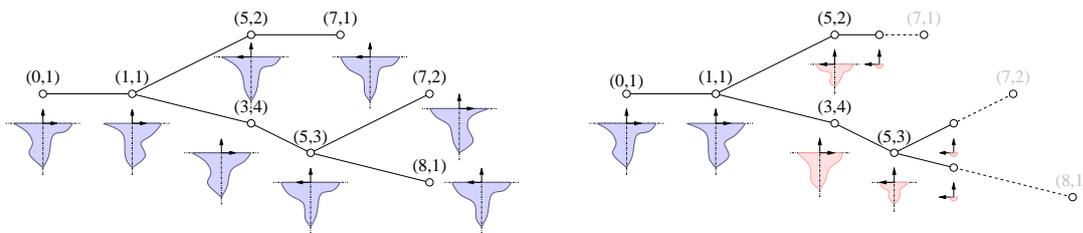


FIG. 5.22 – A gauche : représentation simplifiée d’une fissure générique. A droite : Exemple d’instanciation pour $t = 6$.

Modèle de propagation Notre approche consiste à paramétrer les nœuds \mathcal{N}_i du graphe par un tuple $(t_i, \Delta t_i)$. Le paramètre t_i représente le pas de temps au bout duquel la fissure commence à se former au niveau du nœud \mathcal{N}_i . Le paramètre Δt_i définit la durée nécessaire pour que ce \mathcal{N}_i atteigne sa taille finale (Figure 5.22).

Afin de faciliter l’étape de paramétrisation des fissures génériques, nous avons développé une méthode automatique pour générer les pas de temps au niveaux des nœuds. Sous l’hypothèse que la fissure se propage à une vitesse constante v , et que τ est le temps nécessaire pour

qu'une section se développe entièrement, alors le tuple $(t_j, \Delta t_j)$ peut-être défini par :

$$t_j = t_i + \frac{\delta_{ij}}{v} \quad \Delta t_j = \tau$$

Notre modèle permet de manipuler finement les caractéristiques de la propagation de la fissure dans chaque branche d'une fissure en éditant les paramètres t_i et Δt_i . La formation de l'intégralité d'une sous-branche peut-être retardée ou anticipée en incrémentant ou décrémentant respectivement, t_i par une constante définie par l'utilisateur. De la même manière, la vitesse de formation de la fissure d'un branche peut-être accélérée ou ralentie en diminuant ou augmentant la valeur du paramètre Δt_i . Nous avons développé des outils de haut niveau afin de permettre à l'utilisateur de jouer sur ces contrôles et de définir ainsi avec facilité les caractéristiques de propagation de la fissure de manière globale, ou au niveau de chaque branche.

La création d'une fissure est réalisée en instanciant la fissure générique à un instant t_0 donné. La création de $\mathcal{F}(t_0)$ se fait en deux étapes. Tout d'abord, nous créons le graphe $\mathcal{G}(t_0)$ en ne conservant que les nœuds \mathcal{N}_i tel que $t_i < t_0$. Les arcs \mathcal{A}_{ij} qui vérifient $t_0 > t_j$ sont conservés, tandis que les arcs \mathcal{A}_{ij} tels que $t_i < t_0$ et $t_0 < t_j$ sont décomposés (Figure 5.22). Ensuite, pour les arcs \mathcal{A}_{ij} qui vérifient $t_i < t_0 < t_j$, nous créons un nouveau nœud feuille $\mathcal{N}_k(t_0)$ dans le sous-graphe $\mathcal{G}(t_0)$. L'arc \mathcal{A}_{ik} stocke le même angle $\alpha_{ik} = \alpha_{ij}$ et la distance δ_{ik} est définie par :

$$\delta_{ik}(t_0) = \frac{t_0 - t_i}{t_j - t_i} \delta_{ij}$$

Les profils des nœuds feuilles de $\mathcal{G}(t_0)$ sont définis par des points. Les profils, des nœuds internes, notés \mathcal{P} sont calculés comme suit. Pour chaque nœud \mathcal{N}_i :

1. Si, $t_0 < t_i$, alors la fissure ne s'est pas développée jusqu'au nœud \mathcal{N}_i et dans ce cas, $\mathcal{N}_i(t_0)$ n'est pas ajouté à $\mathcal{G}(t_0)$.
2. Si $t_i < t_0 < t_i + \Delta t_i$, alors le nœud \mathcal{N}_i a commencé à se développer mais n'a pas atteint sa taille finale. Nous définissons les profils de ce nœud par l'homothétie suivante :

$$\mathcal{P}_i(t_0) = \frac{t_0 - t_i}{\Delta t_i} \mathcal{P}_i$$

3. Sinon, $t_0 > t_i + \Delta t_i$ ce qui signifie que le nœud s'est développé jusqu'à sa taille maximale et dans ce cas $\mathcal{N}_i(t_0) = \mathcal{N}_i$.

La figure 5.23 met en scène un personnage de synthèse qui frappe la glace avec son drapeau. La fissure apparaît sous le point d'impact et se propage rapidement à travers la glace. Il a fallu 5 minutes pour créer la fissure statique et 5 minutes supplémentaires pour définir les paramètres de croissance des nœuds de la fissure.

5.4 Conclusion

Dans ce chapitre, nous avons présenté un système pour créer et ajouter rapidement des fissures sur des objets. Notre système permet la création d'une très grande variété de fissures.

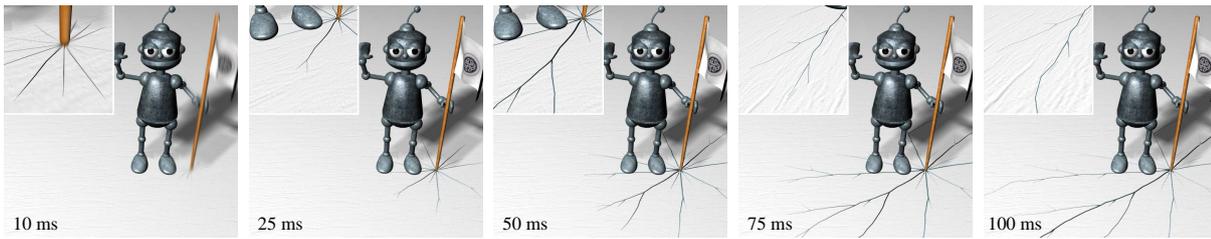


FIG. 5.23 – Animation de la propagation d’une fissure sur la glace.

Le motif de la fissure peut-être créé à partir de photographies ou généré procéduralement. En combinant les deux approches, nous obtenons des fissures dont la forme est physiquement plausible. Notre système fonctionne sur tout type de matériaux dont, le bois, le verre, la terre et la glace. Nous stockons les fissures génériques dans un atlas en fonction de la nature du matériau sur lequel elles peuvent être appliquées. Ceci nous permet de les réutiliser pour générer de nouvelles fissures ou bien, comme base pour de nouveaux motifs.

Nous avons utilisé notre modèle pour créer des fissures et des fractures sur une grande variété d’objets de taille et de nature très différentes : bois, terre, verre, marbre ou encore glace. Toutes les scènes ont été réalisées sur un portable équipé d’un processeur Centrino 1,3Ghz avec 512 Mb de mémoire.

		n	T_o	T_c	T_f	Δ_T
Verre de vin	Fig. 5.14	63	14400	2008	24090	67%
Vase en verre	Fig. 5.13	24	12096	5712	19562	62%
Banc en bois	Fig. 5.16	125	132	3472	4170	3060%
Rampe en bois	Fig. 5.17	638	7728	408	8254	7%
Champignon	Fig. 5.15	63	38160	2160	43830	15%

TAB. 5.1 – Statistiques pour différents modèles fissurés présents dans ce chapitre.

Le tableau 5.1 donne des informations sur le nombre de nœuds n du motif de la fissure, le nombre de triangles de l’objet T_o et de la fissure T_c , le nombre de triangles de l’objet final T_f , et enfin, l’accroissement Δ_T du nombre de triangles engendré par l’ajout de la fissure. Ce que l’on constate, c’est que l’ajout de la fissure n’a pas un coût négligeable. En effet, on peut observer une augmentation du nombre de triangles allant de 7% à 3060%. En fait, plus l’objet initial est détaillé et moins le coût de la fissure est important. Ceci est particulièrement visible dans le cas du banc qui n’est initialement constitué que de 132 triangles et qui avec les fissures est constitué de 4170. En fait, cela est également dû à la différence d’échelle entre la fissure et la planche. Plus la différence est importante et plus le coût d’ajout du détail est important.

Toutefois, bien qu’une texture ait été moins coûteuse, la création de la vraie géométrie de la fissure présente plusieurs intérêts. Tout d’abord, on peut observer sur les figures 5.14 et 5.13 que le fait qu’il y ait la vraie géométrie permet de reproduire les interactions complexes entre

la lumière et le verre fissuré. On peut voir nettement que les réflexions et les réfractions qui ont lieu à l'intérieur du verre augmentent considérablement le réalisme de la scène. Il est très difficile de créer de tels effets par le biais de cartes d'illuminations comme les cartes de variations de normales ou encore les cartes de specularité. De même, dans le cas du jeu d'échec 5.19, le plateau est défini par une texture volumique. Par conséquent, l'intérieur de la fissure passe du blanc au noir en traversant les cases. On peut aussi observer la présence d'auto-ombrage à l'intérieur de la fissure. Tous ces effets ne pourraient être présents si la fissure n'était qu'une texture.

Dans notre modèle, l'objet fissuré est défini par la différence booléenne entre l'objet initial et le volume de la fissure. Or, il n'est pas nécessaire de générer l'objet final pour pouvoir le visualiser. Actuellement, de nombreux travaux portent sur l'évaluation d'un arbre CSG directement par la carte graphique. Cette approche permet ainsi d'éviter les coûts mémoire dus à l'instanciation du modèle. De plus, la fissure elle-même n'est définie que par un squelette en trois dimensions associé à des courbes de profils. Cette représentation permet d'évaluer aisément lorsqu'un point est à l'extérieur ou à l'intérieur de la fissure. Or cette donnée est suffisante pour évaluer l'arbre CSG sur la carte graphique. Dans ce cas, il n'est alors pas nécessaire de générer le volume de la fissure.

Actuellement, le motif de la fissure est représenté par un graphe sans cycle. Comme nous l'avons vu, cette représentation permet déjà de créer un très grand nombre de types de fissures. Toutefois, avec ce modèle, nous ne pouvons pas générer des fissures qui présentent des boucles. La représentation d'une fissure par un graphe avec cycle pose un problème fondamental que nous n'avons pas encore étudié. Ce problème est lié au processus de projection qui n'est pas adapté à une telle structure. Actuellement, lorsqu'une branche subit une déformation locale à cause de la projection, seule sa trajectoire est déviée. Dans le cas d'un graphe avec cycle, il faut pouvoir projeter la boucle en garantissant que la déformation sera minimale sinon, on risque de déformer l'intégralité du motif une fois projeté.

Chapitre 6

Conclusion et perspectives

Dans cette thèse, nous avons proposé une nouvelle approche d'ajout de détails géométriques dans une scène naturelle. Celle-ci repose sur la création d'un atlas et la définition d'un modèle d'évolution et d'un modèle de dispersion. L'atlas est constitué d'un ensemble de fragments géométriques caractéristiques du détail à générer. Le modèle d'évolution permet de combiner ces fragments pour générer les détails directement sur la surface des objets de la scène. Le modèle de dispersion sert à définir les zones dans lesquelles ils peuvent être placés.

Nous avons montré dans les chapitres 3, 4 et 5 que cette approche pouvait être utilisée pour simuler la formation d'éléments de natures très diverses. Nous l'avons appliquée à la simulation de la colonisation de lichens, à la simulation de l'évolution et de la chute des feuilles en automne, et enfin, à la simulation de la formation de fissures.

L'approche que nous avons développée permet de résoudre plusieurs des problèmes liés à l'ajout de détails dans une scène virtuelle : la modélisation précise de détails, le contrôle de la dispersion réaliste des détails dans la scène et la gestion de la mémoire.

Modélisation des détails La première difficulté est de modéliser le détail sur l'objet.

Si l'on prend l'exemple du lichen, il est inconcevable de le modéliser à la main thalle par thalle sur les différents éléments de la scène. L'utilisation d'un atlas de fragments associé à un modèle d'évolution permet de modéliser un grand nombre de détails en combinant des éléments qui sont en principe plus simples à modéliser. Cette approche est similaire à celle utilisée pour la génération de textures à partir de petits échantillons d'images. Le fait de combiner des éléments de petites tailles permet de créer des formes complexes et adaptées à la surface de l'objet sur lequel elles reposent. De plus, plus les fragments sont petits et plus on colle à la surface en minimisant la déformation des fragments.

Contrôle du placement Un autre élément important de notre approche est le modèle de dispersion. Actuellement, de nombreux travaux ont pour objectif de rendre ce travail totalement automatique. Dans la majorité des cas, il s'agit de simulations complexes dont le résultat n'est pas toujours prévisible. Lorsque l'on se place dans le cadre d'un travail artistique, cela est problématique. Au cours de ce travail, nous

nous sommes aperçus qu'il était primordial d'offrir à l'utilisateur deux types d'outils : des outils de haut niveau pour placer grossièrement les détails et des outils de bas niveau pour pouvoir modifier localement le résultat. Cette approche, souvent appelée *global to local*, est très utilisée en modélisation car elle offre un bon contrôle du processus d'ajout de détails. Dans le cas du lichen et des feuilles, la simulation des phénomènes de colonisation et de chute constituait le niveau de contrôle global. L'utilisateur disposait ensuite d'outils simples pour modifier localement le résultat de la simulation. Dans le cas des fissures, nous n'avons utilisé que le niveau local.

Gestion de la mémoire Une autre difficulté inhérente à l'ajout de détails géométriques dans une scène est la gestion de la mémoire. L'utilisation d'un atlas de modèles nous permet de minimiser l'occupation de la scène en mémoire en n'instanciant que les fragments. La forme finale est définie par un processus de référencement. En utilisant des méthodes de quantization, il est possible de réduire le nombre d'éléments de l'atlas et ainsi d'adapter le taux d'utilisation de la mémoire en fonction de la richesse visuelle souhaitée.

L'approche que nous avons proposée dans cette thèse présente cependant des limites qui touchent à la modélisation des détails, le processus d'instanciation et la gestion de l'environnement.

Constitution de l'atlas La première limite de la méthode réside dans la constitution de l'atlas. Nous avons vu que l'utilisation d'un atlas de fragments permet de modéliser simplement les détails de formes complexes en les décomposant en éléments plus simples. Toutefois, tous les détails ne peuvent pas être représentés par un ensemble de fragments et un modèle d'évolution. Tout du moins, pas comme nous l'avons défini. C'est le cas par exemple des lichens filandreux pour lesquels cette approche ne semble pas la plus adaptée.

Instanciation Ensuite, l'utilisation de l'instanciation et du référencement rendent impossible tout changement local de la géométrie des fragments. Par exemple, dans le cas des feuilles, si l'on veut conserver le faible coût en mémoire, il n'est alors pas possible de traiter les feuilles comme des corps mous et ainsi de les déformer localement en fonction du relief ou des feuilles voisines. Toutefois, dès lors que l'application ne nécessite plus de minimiser le coût en mémoire, ceci n'est plus une limitation et l'atlas permet d'instancier chaque élément de la scène.

Environnement Dans notre système, nous avons fait l'hypothèse que les détails ne modifiaient pas localement l'environnement. Cette hypothèse nous permet de ne pas réévaluer les propriétés de l'environnement dans la boucle d'ajout de détails. Toutefois, ceci n'est plausible que dans le cas de petits détails très localisés. Par exemple, l'ajout de lichen joue un rôle très limité dans la modification de l'environnement. A contrario, si on considère l'ajout de feuilles ou même l'ajout de mousse

dans une scène, alors les caractéristiques de l'environnement risquent d'être énormément modifiées localement, notamment en ce qui concerne le taux d'humidité. Dans ce cas, si l'on souhaite plus de réalisme, il faudra inclure l'évaluation de l'environnement dans le processus d'évolution, mais également réévaluer la dispersion. Par conséquent, l'ajout des détails sera plus coûteux en termes de temps de calcul.

Enfin, avant de conclure cette thèse, les points suivants méritent d'être discutés : le problème de la modélisation des fragments, la gestion du niveau de détails et la problématique de la création d'une approche unique.

Modélisation des fragments Dans notre méthode, la principale difficulté de la modélisation des détails repose sur la modélisation des fragments. Une des questions que l'on peut alors se poser est : *Est-il possible de générer les fragments de manière automatique ?* Dans le cas de la synthèse de textures, plusieurs méthodes ont été développées pour générer une nouvelle texture à partir d'échantillons caractéristiques extraits de l'image d'origine. Par analogie, il serait intéressant d'étudier la possibilité d'extraire les fragments à partir d'objets scannés ou de photographies.

Gestion du niveau de détails La gestion du niveau de détails est un élément important lorsque l'on veut pouvoir naviguer en temps réel dans une scène. Dans notre système, les fragments peuvent être générés à la volée et à différents niveaux de détails. Toutefois, notre implémentation actuelle ne permet pas de gérer le niveau de détails de l'élément dans son ensemble. Ainsi, il est possible de dégrader chaque fragment mais pas le détail. De ce fait, il est possible qu'un objet reste coûteux en mémoire alors qu'il est vu de loin. Deux solutions sont à étudier : la première solution consiste à fusionner les fragments en fonction des données de voisinage. Le problème de cette méthode est que le détail risque de pénétrer à l'intérieur de son support, ce qui peut être source d'aliasing. Une autre approche plus intéressante consiste à exploiter le fait que l'on ait la géométrie du détail. Ainsi, il est possible de calculer pour chaque détail la BRDF correspondante et la texture associée. Celles-ci fournissent respectivement les informations d'éclairage et de couleur. Lorsque la distance à laquelle est vu l'objet est trop éloignée pour permettre de discerner les fragments, on remplace la géométrie par la texture et la BRDF associée. Toutefois, ceci n'est possible que si l'on dispose d'une paramétrisation du support.

Vers une approche générique ? Dans cette thèse, nous avons illustré notre approche en développant trois modèles spécifiques pour résoudre les problèmes de la colonisation du lichen, du vieillissement des feuilles et enfin, de la formation de fissures. La définition d'une méthode unifiée permet de combiner facilement les différents phénomènes dans une même scène. Toutefois, le risque de cette méthode est de devoir créer un modèle d'évolution et de dispersion par phénomène ou, dans le meilleur des cas, par catégorie de phénomènes. Il est en effet possible d'utili-

ser l'Open-DLA pour modéliser la formation de givre, ou encore l'Open Markov Chain pour modéliser le vieillissement de pétales. Même dans ces cas pourtant assez proches, il faudra modifier les modèles pour qu'ils tiennent compte de leurs spécificités. L'utilisation d'un atlas associé à un modèle d'évolution et de dispersion permet de représenter une vaste variété d'éléments géométriques. Cette approche peut-être vue comme une sorte de compression. Le détail est constitué par une combinaison des éléments les plus caractéristiques. Sous cette hypothèse, est-il possible de définir un langage de haut niveau permettant de décrire cette combinaison, quel que soit le phénomène que l'on cherche à reproduire ? Une approche dans ce sens serait de définir une grammaire permettant de décrire comment les éléments sont combinés comme dans les travaux sur la phyllotaxie [37].

Publications

Publication dans des revues internationales avec comité de lecture

- [1] Brett Desbenoit, Eric Galin, and Samir Akkouche. Simulating and modeling lichen growth. *Computer Graphics Forum*, 23(3) :341–350, September 2004.
- [2] Brett Desbenoit, Eric Galin, and Samir Akkouche. Modeling cracks and fractures. *The Visual Computer*, 21(8-10) :717–726, 2005.

Publications dans des congrès internationaux avec comité de lecture

- [3] Eric Galin Brett Desbenoit, David Vanderhaghe and Jerome Grosjean. Interactive modeling of mushrooms. In *Proceedings of Eurographics'04, Short Papers*, pages 37–40, 2004.
- [4] Brett Desbenoit, Eric Galin, and Samir Akkouche. Procedural modeling of cracks and fractures. In *Proceedings of Shape Modeling International, Technical Sketch*, pages 346–349, June 2004.
- [5] Brett Desbenoit, Eric Galin, Samir Akkouche, and Jerome Grosjean. Modeling autumn sceneries. In *Proceedings of Eurographics'06 Conférence, Short Papers*, September 2006.

Bibliographie

- [6] D. P. Aikman and A. R. Watkinson. A model for growth and self-thinning in even-aged monocultures of plants. *Ann Bot*, 45(4) :419–427, April 1980.
- [7] M. Alsweis and Oliver Deussen. Modeling and visualization of symmetric and asymmetric plant competition. In *Eurographics Workshop on Natural Phenomena*, pages 83–88, August 2005.
- [8] Norman I. Badler and Welton Becket. Imperfection for realistic image synthesis. *Journal of Visualization and Computer Animation*, 1(1) :26–32, August 1990.
- [9] Farès Belhadj and Pierre Audibert. Modeling landscapes with ridges and rivers : bottom up approach. In *GRAPHITE '05 : Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 447–450, New York, NY, USA, 2005. ACM Press.
- [10] A. Belmonte, H. Eisenberg, and E. Moses. From flutter to tumble : Inertial drag and froude similarity in falling paper. *Physical Review Letters*, 81(2) :345–348, 1998.
- [11] B. Benes. Visual model of plant development with respect to influence of light. In *Computer Animation and Simulation*, pages 125–136, 1997.
- [12] B. Benes and R. Forsbach. Visual simulation of hydraulic erosion. In V. Skala, editor, *Journal of WSCG*, volume 10, 2002.
- [13] Bedrich Benes. An efficient estimation of light in simulation of plant development. In *EGCAS '96 : Seventh International Workshop on Computer Animation and Simulation*, 1996.
- [14] Bedrich Benes. A stable modeling of large plant ecosystems. In *International Conference on Computer Vision and Graphics Zakopane*, pages 94–101, 2002.
- [15] Bedrich Benes, Javier Abdul Córdoba, and Juan Miguel Soto. Interacting agents with memory in virtual ecosystems. In *WSCG*, 2003.
- [16] L. Bernd and O. Deussen. Interactive modelling of plants, 1999.
- [17] Jules Bloomenthal. Modeling the mighty maple. In *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 305–311, New York, NY, USA, 1985. ACM Press.

- [18] Carles Bosch, Xavier Pueyo, Stéphane Mérillou, and Djamchid Ghazanfarpour. A physically-based model for rendering realistic scratches. *Computer Graphics Forum*, 23(3) :361–370, September 2004.
- [19] Frédéric Boudon, Przemyslaw Prusinkiewicz, Pavol Federl, Christophe Godin, and Radoslaw Karwowski. Interactive design of bonsai tree models. *Computer Graphics Forum*, 22(3) :591–600, September 2003.
- [20] Michael Braitmaier, Joachim Diepstraten, and Thomas Ertl. Real-time rendering of seasonal influenced trees. In *Theory and Practice of Computer Graphics*, 2004.
- [21] John W. Buchanan and Paul Lalonde. An observational model for illuminating isolated scratches.
- [22] Yao-Xun Chang and Zen-Chung Shih. The synthesis of rust in seawater. *The Visual Computer*, 19(1) :50–66, 2003.
- [23] Yanyun Chen, Lin Xia, Tien-Tsin Wong, Xin Tong, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Visual simulation of weathering by gamma-ton tracing. *ACM Transactions on Graphics*, 24(3) :1127–1133, August 2005.
- [24] Norishige Chiba, Kazunobu Muraoka, Akio Doi, and Junya Hosokawa. Rendering of forest scenery using 3d textures. *The Journal of Visualization and Computer Animation*, 8(4) :191–199, 1997.
- [25] Michael F. Cohen, Jonathan Shade, Stefan Hiller, and Oliver Deussen. Wang tiles for image and texture generation. *ACM Trans. Graph.*, 22(3) :287–294, 2003.
- [26] Phillippe de Reffye, Claude Edelin, Jean Francon, Marc Jaeger, and Claude Puech. Plant models faithful to botanical structure and development. *Computer Graphics (Proceedings of SIGGRAPH 88)*, 22(4) :151–158, August 1988.
- [27] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series*, pages 317–324, August 1999.
- [28] Oliver Deussen, David S. Ebert, Ron Fedkiw, F. Kenton Musgrave, Przemyslaw Prusinkiewicz, Doug Roble, Jos Stam, and Jerry Tessendorf. The elements of nature : interactive and realistic techniques. In *GRAPH '04 : Proceedings of the conference on SIGGRAPH 2004 course notes*, page 32, New York, NY, USA, 2004. ACM Press.
- [29] Oliver Deussen, Patrick M. Hanrahan, Bernd Lintermann, Radomír Měch, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *Proceedings of SIGGRAPH*, pages 275–286, 1998.
- [30] Oliver Deussen and Bernd Lintermann. A modelling method and user interface for creating plants. In Wayne A. Davis, Marilyn Mantei, and R. Victor Klassen, editors, *Graphics Interface '97*, pages 189–198. Canadian Human-Computer Communications Society, 1997.

-
- [31] Julie Dorsey, Alan Edelman, Justin Legakis, Henrik Wann Jensen, and Hans K hling Pedersen. Modeling and rendering of weathered stone. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 225–234, August 1999.
- [32] Julie Dorsey and Patrick M. Hanrahan. Modeling and rendering of metallic patinas. In *Proceedings of SIGGRAPH*, pages 387–396, 1996.
- [33] Julie Dorsey, Hans K hling Pedersen, and Patrick M. Hanrahan. Flow and changes in appearance. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pages 411–420, August 1996.
- [34] P. Federl and P. Prusinkiewicz. Modelling fracture formation in bi-layered materials, with applications to tree bark and drying mud. In *Proceedings of Western Computer Graphics Symposium*, pages 29–35, 2002.
- [35] P. Federl and P. Prusinkiewicz. Finite element model of fracture formation on growing surfaces. *Lecture Notes in Computer Science*, 3037 :138–145, Jan 2004.
- [36] Alain Fournier, Don Fussell, and Loren Carpenter. Computer rendering of stochastic models. *Commun. ACM*, 25(6) :371–384, 1982.
- [37] Deborah R. Fowler, Przemyslaw Prusinkiewicz, and Johannes Battjes. A collision-based model of spiral phyllotaxis. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 361–368. ACM Press, 1992.
- [38] Chi-Wing Fu and Man-Kang Leung. Texture tiling on arbitrary topological surfaces using wang tiles. In *Rendering Techniques 2005 : 16th Eurographics Workshop on Rendering*, pages 99–104, June 2005.
- [39] S. Gobron and N. Chiba. Simulation of peeling using 3d-surface cellular automata. In *9th Pacific Conference on Computer Graphics and Applications*, pages 338–347, October 2001.
- [40] St phane Gobron and Norishige Chiba. Corrosion : Simulating and rendering. In *IPSIJ-Tohoku workshop, Morioka, Japan*, pages 3–9, 1997.
- [41] St phane Gobron and Norishige Chiba. Crack pattern simulation based on 3d surface cellular automata. *The Visual Computer*, 17(5) :287–309, 2001.
- [42] Mark S. Hammel, Przemyslaw Prusinkiewicz, and Brian Wyvill. Modelling compound leaves using implicit contours. In *CG International '92 : Proceedings of the 10th International Conference of the Computer Graphics Society on Visual computing : integrating computer graphics with computer vision*, pages 199–212, New York, NY, USA, 1992. Springer-Verlag New York, Inc.
- [43] J. Hart and B. Baker. Implicit modeling of tree surfaces. In *In Implicit Surfaces '96*, pages 143–152, 1996.
- [44] Koichi Hirota, Yasuyuki Tanoue, and Toyohisa Kaneko. Generation of crack patterns with a physical model. *The Visual Computer*, 14(3) :126–137, 1998.

- [45] Koichi Hirota, Yasuyuki Tanoue, and Toyohisa Kaneko. Simulation of three-dimensional cracks. *The Visual Computer*, 16(7) :371–378, 2000.
- [46] Doug L. James and Dinesh K. Pai. BD-Tree : Output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3), August 2004.
- [47] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics*, 22(3) :943–949, July 2003.
- [48] Alexander Keller and Wolfgang Heidrich. Interleaved sampling. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 269–276, London, UK, 2001. Springer-Verlag.
- [49] Alexander Keller and Wolfgang Heidrich. Interleaved sampling. In *Rendering Techniques : 12th Eurographics Workshop on Rendering*, pages 269–276, 2001.
- [50] Brendan Lane and Przemyslaw Prusinkiewicz. Generating spatial distributions for multilevel models of plant communities. In *Proceedings of Graphics Interface*, pages 69–80, 2002.
- [51] Sylvain Lefebvre and Fabrice Neyret. Synthesizing bark. In *Rendering Techniques 2002 : 13th Eurographics Workshop on Rendering*, pages 105–116, June 2002.
- [52] J. P. Lewis. Generalized stochastic subdivision. *ACM Transactions on Graphics*, 6(3) :167–190, July 1987.
- [53] Fuhrer M., Jensen H. W., and Prusinkiewicz P. Modeling hairy plants. *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 217–226, 2004.
- [54] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, August 1982.
- [55] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freedman and Co., New York, 1983.
- [56] Karl Maritaud. *Rendu réaliste d'arbres vus de près en images de synthèse*. PhD thesis, University of Limoges, France, December 2003.
- [57] S. Merillou, Jean-Michel Dischler, and D. Ghazanfarpour. Surface scratches : measuring, modeling and rendering. *The Visual Computer*, 17(1) :30–45, 2001.
- [58] Stephane Merillou, Jean-Michel Dischler, and Djamchid Ghazanfarpour. Corrosion : Simulating and rendering. In *Graphics Interface*, pages 167–174, 2001.
- [59] Gavin Miller. Efficient algorithms for local and global accessibility shading. In *Proceedings of SIGGRAPH*, pages 319–326, 1994.
- [60] Gavin S. P. Miller. The definition and rendering of terrain maps. *Computer Graphics (Proceedings of SIGGRAPH 86)*, 20(4) :39–48, August 1986.
- [61] S. Mochizuki, D. Cai, T. Komori, H. Kimura, and R. Hori. Virtual autumn coloring system based on biological and fractal model. In *Proceedings of Pacific Graphics*, pages 348–354, 2001.

-
- [62] Neil Molino, Zhaosheng Bao, and Ron Fedkiw. A virtual node algorithm for changing mesh topology during simulation. *ACM Transactions on Graphics*, 23(3) :385–392, August 2004.
- [63] Matthias Müller, Leonard McMillan, Julie Dorsey, and Robert Jagnow. Real-time simulation of deformation and fracture of stiff materials. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 113–124, New York, NY, USA, 2001. Springer-Verlag New York, Inc.
- [64] Lars Mündermann, Peter MacMurchy, Juraj Pivovarov, and Przemyslaw Prusinkiewicz. Modeling lobed leaves. In *Computer Graphics International*, pages 60–67, 2003.
- [65] F. Kenton Musgrave, Craig E. Kolb, and Robert S. Mace. The synthesis and rendering of eroded fractal terrains. *Computer Graphics (Proceedings of SIGGRAPH 89)*, 23(3) :41–50, July 1989.
- [66] Kenji Nagashima. Computer generation of eroded valley and mountain terrains. *The Visual Computer*, 13(9-10) :456–464, 1998.
- [67] Fabrice Neyret and Marie-Paule Cani. Pattern-based texturing revisited. In *SIGGRAPH 99 Conference Proceedings*, pages 235–242. ACM SIGGRAPH, Addison Wesley, August 1999.
- [68] Hiraku Nishimori and Noriyuki Ouchi. Formation of ripple patterns and dunes by wind blown sand. *Physical Review Letters*, 71(1) :197–200, 1993.
- [69] Alan Norton, Greg Turk, Bob Bacon, John Gerth, and Paula Sweeney. Animation of fracture by physical modeling. *Vis. Comput.*, 7(4) :210–219, 1991.
- [70] James F. O’Brien, Adam W. Bargteil, and Jessica K. Hodgins. Graphical modeling and animation of ductile fracture. *ACM Transactions on Graphics*, pages 291–294, July 2002.
- [71] James F. O’Brien and Jessica K. Hodgins. Graphical modeling and animation of brittle fracture. In *Proceedings of SIGGRAPH*, pages 137–146, 1999.
- [72] E. Paquette, P. Poulin, and G. Drettakis. Surface aging by impacts. In *Graphics Interface*, 2001.
- [73] Eric Paquette, Pierre Poulin, and George Drettakis. The simulation of paint cracking and peeling. In *Graphics Interface 2002*, pages 59–68, May 2002.
- [74] Heinz-Otto Peitgen and Dietmar Saupe, editors. *The Science of Fractal Images*. Springer-Verlag New York, Inc., New York, NY, USA, 1988.
- [75] P. Prusinkiewicz, A. Lindenmayer, and J. Hanan. Developmental models of herbaceous plants for computer imagery purposes. In *ACM SIGGRAPH*, volume 22, Atlanta, Georgia, August 1988. ACM SIGGRAPH.
- [76] Przemyslaw Prusinkiewicz and Mark Hammel. A fractal model of mountains with rivers. In *Graphics Interface ’93*, pages 174–180, May 1993.

- [77] Przemyslaw Prusinkiewicz, Mark James, and Radomír Měch. Synthetic topiary. In *Proceedings of SIGGRAPH*, pages 351–358, 1994.
- [78] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.
- [79] Przemyslaw Prusinkiewicz, Lars Mündermann, Radoslaw Karwowski, and Brendan Lane. The use of positional information in the modeling of plants. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 289–300, August 2001.
- [80] William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 313–322, New York, NY, USA, 1985. ACM Press.
- [81] Adam Runions, Martin Fuhrer, Brendan Lane, Pavol Federl, Anne-Gaëlle Rolland-Lagan, and Przemyslaw Prusinkiewicz. Modeling and visualization of leaf venation patterns. *ACM Trans. Graph.*, 24(3) :702–711, 2005.
- [82] Jeffrey Smith, Andrew Witkin, and David Baraff. Fast and controllable simulation of the shattering of brittle objects. *Computer Graphics Forum*, 20(2) :81–91, 2001.
- [83] Cyril Soler, François Sillion, Frédéric Blaise, and Philippe Dereffye. An efficient instantiation algorithm for simulating radiant energy transfer in plant models. *ACM Transactions On Graphics*, 22(2), April 2003.
- [84] Robert Walker Sumner. Pattern formation in lichen, 2001. Ph.D. thesis, *Département of Electrical Engineering and Computer Science*, Massachusetts institute of technology, 2001.
- [85] Y. Tanabe and K. Kaneko. Behavior of a falling paper. *Physical Reveiew Letters*, 73(10) :1372–1375, 1994.
- [86] Demetri Terzopoulos and Kurt Fleischer. Deformable models. *The Visual Computer*, 4(6) :306–331, December 1988.
- [87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Computer Graphics (Proceedings of SIGGRAPH 87)*, volume 21, pages 205–214, July 1987.
- [88] X. Viennot, G. Eyrolles, N. Janey, and D. Arquès. Combinatorial analysis of ramified patterns and computer imagery of trees. *SIGGRAPH Comput. Graph.*, 23(3) :31–40, 1989.
- [89] R. F. Voss. Random fractal forgeries. In *R. A. Earnshaw, editor, Fundamental Algorithms in Computer Graphics*, pages 805–835, Berlin, 1985. Springer-Verlag.
- [90] Xi Wang, Lifeng Wang, Ligang Liu, Shi-Min Hu, and Baining Guo. Interactive modeling of tree bark. In *Pacific Conference on Computer Graphics and Applications*, pages 83–91, October 2003.

-
- [91] Jason Weber and Joseph Penn. Creation and rendering of realistic trees. In *Proceedings of SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series*, pages 119–128, August 1995.
- [92] T.A. Witten and L.M. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. *Physical Review Letters*, 47 :1400–1403, 1981.
- [93] Tien-Tsin Wong, Wai-Yin Ng, and Pheng-Ann Heng. A geometry dependent texture generation framework for simulating surface imperfections. In *Eurographics Rendering Workshop*, pages 139–150, 1997.
- [94] Brian Wyvill, Kees van Overveld, and Sheelagh Carpendale. Rendering cracks in batik. In *NPAR 2004*, pages 61–70, June 2004.

TITRE : Modélisation et simulation des scènes naturelles complexes

RESUME

Dans cette thèse, nous abordons le problème de la modélisation de scènes naturelles réalistes. Nous nous sommes plus particulièrement intéressés à la modélisation des détails et à leur répartition. Ces deux aspects jouent un rôle fondamental sur le réalisme des scènes virtuelles. L'approche que nous proposons dans cette thèse repose sur la définition de trois éléments clés : un atlas de formes élémentaires, un algorithme de dispersion et un algorithme de propagation. L'atlas est composé des briques de bases utilisées pour la génération de la géométrie des détails. L'algorithme de propagation, quant à lui, décrit comment ces éléments vont se combiner pour former l'objet final. Enfin, l'algorithme de dispersion définit leur répartition dans la scène. L'approche que nous décrivons permet également de prendre en compte les caractéristiques de l'environnement lors des étapes de placement et de propagation. Nous avons appliqué cette approche à trois éléments que l'on rencontre couramment dans la nature : les lichens, les feuilles et les fissures.

Mots-clés : modélisation, simulation, détails, scène naturelle.

TITLE : Modelling and simulating complexe natural scenes

ABSTRACT

In this thesis, we address the modelling realistic natural sceneries. We focus on the modeling of details and their distribution in a complex scene. Details have a great impact over the overall realism of the final scene. Our approach consists in modeling details such as mushrooms, lichens, leaves or fractures and storing them in an atlas of shapes. Those details are distributed in the scene according to specific physically and biologically based dispersion and propagation algorithms, which are controlled by the parameters of the environment such as the wetness, the temperature or the amount of direct and indirect lighting. Our approach enables us to add a vast variety of details without the burden of editing them by hand.

Keywords : modelling, simulating, details, natural scene.

DISCIPLINE : Informatique

INTITULE ET ADRESSE DU LABORATOIRE :

LIRIS UMR 5205
Université Claude Bernard
Bâtiment Nautibus (710),
43, Boulevard du 11 Novembre 1918
69622 VILLEURBANNE CEDEX

