



Génération de détails dans les mondes procéduraux

François Grosbellet

► To cite this version:

François Grosbellet. Génération de détails dans les mondes procéduraux. Modélisation et simulation. Université de Limoges, 2015. Français. NNT : . tel-01289869v1

HAL Id: tel-01289869

<https://hal.science/tel-01289869v1>

Submitted on 17 Mar 2016 (v1), last revised 29 Feb 2016 (v2)

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

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

Thèse
pour obtenir le grade de
Docteur de l'Université de Limoges
en informatique

réalisée sous la co-direction
de Stéphane Mérillou et de Éric Galin

Génération de détails dans les mondes procéduraux

François GROSBELLET

Thèse soutenue le : 20 novembre 2015

Composition du Jury

Rapporteurs :

Jean Jacques BOURDIN	Professeur, Université Paris 8
Christophe RENAUD	Professeur, Université du Littoral Côte d'Opale

Examineurs :

Jean-Michel DISCHLER	Professeur, Université de Strasbourg
Guillaume GILET	Maitre de conférences, Université de Limoges

Directeurs :

Stéphane MÉRILLOU	Professeur, Université de Limoges
Éric GALIN	Professeur, Université Lumière Lyon 2



Remerciements

Je remercie d'abord mes directeurs, Stéphane Mérillou et Éric Galin, de m'avoir proposé de faire cette thèse sous leur direction et de m'avoir encadré. Venant d'un master professionnel, sans la proposition de thèse que vous m'avez faites, je n'aurais très certainement jamais découvert le monde de la recherche. Merci pour la qualité du travail, des conseils et du soutien que vous m'avez apporté au long des quatre années de cette thèse. Vous avez également toujours cherché à me faire gagner du temps sur l'aspect administratif parfois complexe lié au co-encadrement, ce que j'ai beaucoup apprécié.

Je remercie également Adrien Peytavie et Éric Guérin qui, même s'ils n'étaient pas officiellement encadrants, se sont investis comme si cela était le cas. Merci pour votre investissement et pour le temps passer à travailler ensemble, pour les solutions que vous avez contribué à apporter, pour tous les échanges dont j'ai énormément appris.

Merci à Jean Jacques Bourdin et Christophe Renaud d'avoir accepté d'être rapporteurs sur cette thèse, de m'avoir consacré une partie non négligeable de votre temps, et de m'avoir apporté votre expertise à travers les retours sur ce manuscrit. Merci à Jean-Michel Dischler et à Guillaume Gilet d'avoir accepté d'être mes examinateurs. Dans l'ensemble, merci au membres du jury pour votre investissement.

Merci aux différentes générations de doctorants qui se sont succédés dans le bureau LIRIS 1 : Jérémy, Jean-David, Abdoulaye, Grégoire, Jérémy bis, Matthieu. Merci aux autres doctorants du LIRIS : Hélène, Marie-Neige, Karolina, Joseph. Merci pour les idées, les séances de travail collectif en général et de débogage en particulier, le soutien et la motivation. Merci pour les pauses AML, pour les cookies et les gâteaux, pour les empilements de pneus, pour les duels au Nerf, pour les batailles à coups de Kirby, pour ces moments passé à faire et à raconter n'importe quoi, qui sont nécessaires pour faire une pause, décompresser et repartir de plus belle.

Merci à Dodolf, Mierre, Karganys et TFC pour les discussions concernant le travail, pour les soirées rétro-biering, et bien entendu pour tout le reste, que je n'ai pas pas la place d'énumérer ici.

Merci à David Coeurjolly pour les discussions, les avis, les relectures et les différents coups de mains.

Merci à Guillaume Gillet, pour, en plus de ta présence dans mon jury, le temps passé à travailler et à échanger sur la recherche, pour les demandes d'hébergement lors des déplacements sur Limoges pour lesquelles tu as toujours répondu présent, pour le temps passé ensemble à discuter et à échanger en différentes occasions.

Un grand merci à ma famille pour le soutien et les encouragements inconditionnels.

Et finalement, merci à Anaïs pour ta présence, ton écoute, tes conseils et ton soutien, particulièrement pendant la période de rédaction qui fût loin d'être évidente.

Table des matières

Introduction	5
1 État de l'art	7
1 Introduction	8
2 Génération procédurale de contenu	10
2.1 Terrains	10
2.2 Végétation	12
2.3 Routes et réseaux routiers	13
2.4 Villes	14
2.5 Bâtiments	16
2.6 Synthèse	17
3 Vieillissement et changements d'apparence	17
3.1 Dommages mécaniques	19
3.1.1 Fractures	19
3.1.2 Impacts, chocs et rayures	19
3.1.3 Transport de matière	19
3.1.4 Poussières	20
3.2 Attaques biologiques	20
3.3 Attaques chimiques	21
3.4 Synthèse	21
4 Ajouts de géométrie	22
4.1 Phénomènes hivernaux	22
4.1.1 Génération de neige	22
4.1.2 Génération de glace	29
4.1.3 Synthèse	32
4.2 Phénomènes végétaux	32
4.2.1 Feuilles	32
4.2.2 Herbe	34
5 Conclusion	39

2	Objets environnementaux	40
1	Introduction	41
2	Arbre de construction	41
2.1	Objets environnementaux	43
2.1.1	Objet environnemental statique	43
2.1.2	Objets environnementaux paramétrés	44
2.2	Opérateurs de combinaison	46
3	Évaluation des effets	47
3.1	Réaction à l'environnement	47
3.2	Interactions	49
4	Création des scènes	51
4.1	Base de données d'objets environnementaux	51
4.2	Création de l'arbre de construction	52
4.2.1	Édition interactive	52
4.2.2	Génération procédurale	53
4.3	Génération de la scène décorée finale	54
5	Conclusion	54
3	Environnement	56
1	Introduction	57
2	Définition de l'environnement	57
3	Les champs d'influence	58
3.1	Spécificités des champs d'herbe et de feuilles	59
3.2	Primitives à squelette	59
3.2.1	Point	59
3.2.2	Segment	60
3.2.3	Courbe	60
3.2.4	Cylindre	61
3.2.5	Boîte	61
3.2.6	Comparaison des temps de calcul	61
3.3	Primitives à base d'image	62
4	Combinaison des champs d'influence	62
4.1	Cas général	63
4.2	Cas particulier de l'accessibilité	63
5	Accélérations	66
6	Conclusion	68
4	Effets	69
1	Introduction	70
2	Neige	70
2.1	Définition des primitives de neige	70
2.2	Fonction d'élévation de neige	71
2.3	Catégories de primitives de neige	73
2.3.1	Primitives de neige géométriques	73
2.3.2	Primitives de neige procédurales	74
2.3.3	Comparaison entre primitives géométriques et procédurales	78
2.4	Résultats et conclusion	79
3	Glace	81
3.1	Définition des primitives de glace	81
3.2	Sélection, paramétrisation et création du modèle géométrique	82

	3.2.1	Sélection	82
	3.2.2	Détermination des paramètres	82
	3.2.3	Création du modèle géométrique	83
	3.3	Résultats et conclusion	84
4	Herbe		86
	4.1	Définition des primitives d’herbe	86
	4.2	Les différentes primitives d’herbe	86
	4.3	Sélection et paramétrisation des touffes d’herbes candidates	89
	4.3.1	Instanciation	89
	4.3.2	Détermination des paramètres	90
	4.4	Résultats et conclusion	92
5	Feuilles		94
	5.1	Définition des primitives de feuilles	94
	5.2	Les différentes primitives de feuilles	94
	5.3	Distribution des feuilles candidates	96
	5.4	Sélection et paramétrisation des feuilles candidates	98
	5.4.1	Sélection	98
	5.4.2	Détermination des paramètres	98
	5.5	Résultats et conclusion	99
6	Accélérations		102
7	Conclusion		103
Conclusion et perspectives			105
A Objets environnementaux paramétrés			107

Introduction

Les mondes virtuels sont de plus en plus importants au sein de la société humaine. Depuis plus de quarante ans, les technologies de génération de mondes virtuels sont en perpétuelle évolution, et cette tendance s'est accélérée au cours des quinze premières années de ce siècle. Autrefois réservés à une industrie du divertissement presque anecdotique et à des niches professionnelles très spécifiques, les mondes virtuels sont dorénavant exploités dans de nombreux pans de la société. L'industrie du divertissement a énormément changé en quelques années. Un nombre extrêmement important de productions cinématographiques a massivement recours aux effets spéciaux et aux images de synthèses, délaissant par la même occasion les techniques de prise de vue traditionnelles. L'industrie du jeu vidéo s'est énormément développée dans la dernière décennie grâce à l'introduction sur le marché de machines puissantes et grâce à l'avènement des smartphones, devenant la première industrie de loisir sur le plan économique. Les mondes virtuels sont actuellement utilisés dans des cadres professionnels extrêmement variés : formations professionnelles de tous types par le biais de simulateurs, assistance aux praticiens dans les milieux médicaux, systèmes d'informations géographiques pour la planification urbaine, la lutte contre les incendies ou simplement comme évolution logique des cartes géographiques au format papier, etc. Les applications des mondes virtuels sont extrêmement vastes, de plus en plus intégrées dans la vie courante, et de nouvelles façons de les exploiter émergent chaque jour. Après plusieurs années où l'on a pu observer la façon dont ces technologies ont été assimilées et utilisées, on dispose aujourd'hui du recul nécessaire pour affirmer avec certitude que les besoins en termes de mondes virtuels vont continuer à croître rapidement.

La génération de mondes virtuels est un domaine scientifique très vaste qui est plus que jamais, même après des années de recherches intensives sur le sujet, au centre de problématiques très variées. Plusieurs aspects sont à prendre en compte pour qu'un monde virtuel, qu'il soit conçu pour être réaliste ou non, soit convaincant : il doit être suffisamment grand pour que l'on n'en perçoive pas les limites, fournir assez de diversité pour que l'on ne puisse pas y déceler de répétitions, et posséder des détails visibles au moins jusqu'au point de vue de l'observateur. L'amélioration et la recrudescence des techniques de génération de contenu, et notamment des techniques de génération procédurale, ont permis de résoudre en partie ces difficultés, en générant automatiquement et rapidement des quantités croissantes de contenu dont la qualité est en constante amélioration. Cependant, de très nombreuses pistes peuvent et doivent encore être explorées, et alors que des verrous scientifiques et technologiques sont petit à petit résolus, les limites des mondes virtuels sont sans cesse repoussées. Les quantités et la qualité du contenu que l'on veut générer ne cessent de croître, rendant progressivement les méthodes existantes obsolètes et créant continuellement de nouveaux besoins et de nouveaux problèmes à résoudre.

Les méthodes existantes sont soit conçues pour générer du contenu à une très grande échelle en possédant une précision très faible (e. g. générer de la neige sur de grands terrains avec des détails de l'ordre du mètre), soient conçues pour générer du contenu à une très petite échelle avec des détails très précis (e. g. modéliser de la neige sur des objets dont la taille est

de l'ordre du mètre avec une précision inférieure au millimètre). Il n'existe pas de méthode fonctionnant à la fois sur une grande échelle et avec une précision élevée. Ces méthodes sont de plus majoritairement axées sur la production d'un résultat et ne prennent pas réellement en compte la position de l'utilisateur dans leur fonctionnement, ce qui les rends difficilement contrôlables.

Dans cette thèse, nous nous intéressons à ces deux problématiques. Notre but est de concevoir une méthode capable de générer automatiquement du contenu très détaillé sur de grandes échelles. Nous voulons de plus mettre l'utilisateur au centre de la méthode pour lui permettre de contrôler précisément et intuitivement les résultats produits.

Le chapitre 1 contient un état de l'art présentant trois catégories de méthodes de génération de contenu graphique : les méthodes de génération procédurale, les méthodes de vieillissement et de changements d'apparence, et les méthodes d'ajout de géométrie. Ces catégories regroupent des méthodes dont les objectifs sont très distincts. Les méthodes de génération procédurale sont conçues pour créer automatiquement un vaste panel de types de contenu en fonction de règles et de paramètres. Elles prennent la forme d'outils puissants, automatisant des tâches fastidieuses et contraignantes, offrant la possibilité de générer rapidement soit de grandes quantités de contenu, soit du contenu très détaillé. Les techniques de vieillissement et de changement d'apparence ne génèrent pas le contenu graphique en lui même mais différentes versions altérées du contenu qui leur est fourni. Elles reproduisent fidèlement le fonctionnement de phénomènes du monde réel (tâches, rouilles, etc.) dans le but de fournir du contenu aussi réaliste que possible. Les méthodes d'ajout de géométrie visent également à créer de la diversité à partir du contenu qui leur est fourni, mais contrairement à la famille de méthodes précédente, elles ne modifient pas intrinsèquement ce contenu : elles génèrent des détails supplémentaires, aussi appelés *décorations géométriques*, qui enrichissent ce contenu, mais qui peuvent en être décorrélées.

Nous présentons ensuite dans le chapitre 2 un nouveau modèle pour la génération de grandes scènes contenant de grandes quantités de détails. Ce modèle repose sur l'utilisation conjointe d'une nouvelle catégorie d'objets que nous avons baptisés *objets environnementaux* et de champs scalaires, organisés dans un arbre de construction. Les objets environnementaux sont des objets super spécialisés qui génèrent eux même leurs décorations géométriques afin de produire automatiquement des détails et de la diversité. Nous utilisons les champs scalaires pour définir les différentes informations qui guident le processus de génération des décorations des objets environnementaux.

Nous examinons en détail le fonctionnement des champs scalaires, que nous regroupons sous le concept d'*environnement*, dans le chapitre 3. L'environnement est une entité abstraite qui définit différentes informations, telles que les précipitations de neige ou la température, en tout point de la scène. Cet ensemble d'informations est utilisé par les objets environnementaux afin de guider le processus de création des décorations. Nous présentons également une méthode d'accélération pour diminuer les temps de calculs nécessaires à l'évaluation de ces informations.

Finalement, nous proposons dans le chapitre 4 quatre méthodes pour générer les décorations géométriques des objets environnementaux : la neige, la glace, l'herbe et les feuilles. L'objectif de ces méthodes procédurales est la génération automatique et en temps interactif de grandes quantités de détails en offrant un maximum de contrôle à l'utilisateur, tout en exigeant un minimum d'interventions de sa part. La dernière partie du chapitre est consacrée à une technique d'accélération pour ces méthodes.

Chapitre 1

État de l’art

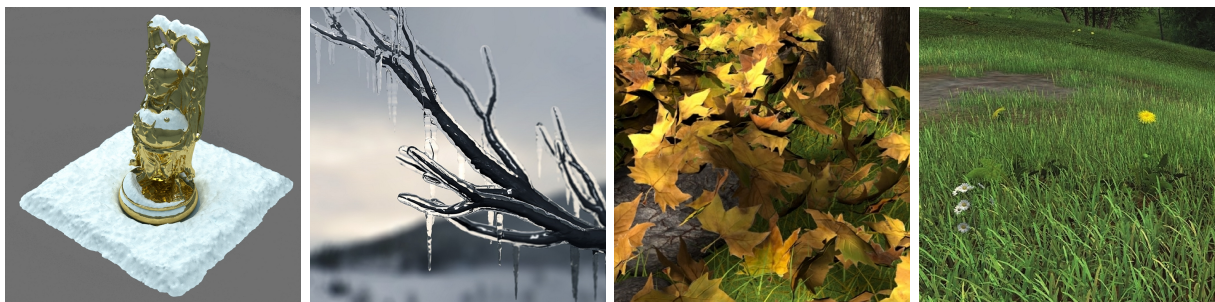


FIGURE 1.1 – Génération de détails par des méthodes d’ajouts de géométrie. Les images présentées dans cette figure illustrent les meilleurs résultats que les méthodes existantes permettent de produire. De gauche à droite : manteau neigeux modélisé à partir de *levels-sets* [60], stalactites générées avec un modèle physique [48], distribution de feuilles par trajectoires procédurales [36], et herbe distribuée par carte de densité et rendu en temps réel par éclairage dynamique [20].

Table des matières

1	Introduction	8
2	Génération procédurale de contenu	10
2.1	Terrains	10
2.2	Végétation	12
2.3	Routes et réseaux routiers	13
2.4	Villes	14
2.5	Bâtiments	16
2.6	Synthèse	17
3	Vieillessement et changements d’apparence	17
3.1	Domages mécaniques	19
3.2	Attaques biologiques	20
3.3	Attaques chimiques	21
3.4	Synthèse	21
4	Ajouts de géométrie	22
4.1	Phénomènes hivernaux	22
4.2	Phénomènes végétaux	32
5	Conclusion	39

1 Introduction

Il existe une grande quantité de méthodes permettant de générer des mondes virtuels peuplés de contenu très diversifié : des terrains vastes et détaillés, des forêts constituées de milliers de végétaux, des réseaux routiers et des routes connectant des villes procédurales comportant des milliers de bâtiments dont les modèles géométriques sont très précis, etc. Le problème le plus fréquemment rencontré lors de l'utilisation de ces méthodes est le manque de réalisme des objets générés. Pour que ces objets soient convaincants, il faut que chacun des éléments qui les compose le soit également. Si les bâtiments qui constituent une ville se ressemblent tous ou s'ils donnent tous l'impression d'être neufs et ne portent pas de traces liées à leur utilisation, au temps ou à différents phénomènes, le résultat obtenu manquera grandement de réalisme : l'apparence de la ville générée sera très artificielle. De la même manière, pour qu'une forêt soit crédible, elle doit comporter une grande quantité de caractéristiques visuelles essentielles. Ses arbres doivent varier en âge, en taille et en essence, on doit y trouver des arbres morts, colonisés par la mousse et le lierre, abimés par le vent, etc. La clé de l'augmentation du réalisme réside dans des méthodes puissantes capables de produire un type de contenu dont l'aspect peut varier, ou dans des méthodes destinées à faire varier l'aspect de contenu existant.

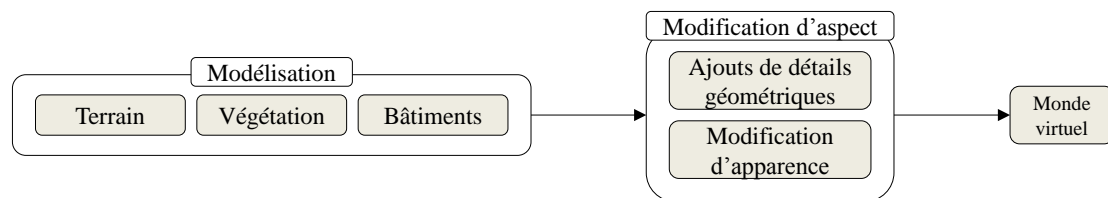


FIGURE 1.2 – Processus de création des mondes virtuels.

La création de mondes virtuels se fait en deux étapes. La première étape consiste à générer du contenu par étapes successives et la seconde à faire varier ce contenu en fonction des besoins (Figure 1.2). Dans cet état de l'art, nous nous intéresserons principalement à la seconde étape. Dans le monde réel, les modifications d'aspect ont des causes multiples (Figure 1.3). Les chocs et les coups enfoncent la surface des objets et peuvent détériorer les couches extérieures telles que la peinture ou le vernis. Les métaux s'oxydent, ce qui modifie leur apparence et leur structure. Des minéraux cristallisent à la surface des objets qui sont exposés à la pollution. La végétation peut se développer à la surface des objets, en la recouvrant et en la détériorant. Le changement de phase de l'eau qui ruisselle crée des stalactites. De nombreux phénomènes, naturels ou non, créent des dépôts à la surface des objets : neige, feuilles mortes, poussière, etc.



FIGURE 1.3 – Exemples réels de variations d'apparences. De gauche à droite : rayures causées par un choc mécanique, formation de rouille due à la corrosion du métal, formation de gypse causée par la pollution, croissance de mousse sur un mur, croissance d'herbe entre des pavés, formation de stalactites au bord d'un toit, dépôt de feuilles mortes, couche de neige.

Deux catégories de causes font varier l'aspect des objets : soit en modifiant leurs apparences, soit en leur ajoutant des détails géométriques (Figure 1.4). Dans ce contexte, modifier l'apparence d'un objet signifie modifier sa géométrie ou son matériau. Un objet métallique se patinera par exemple avec le temps et changera partiellement de couleur. Si cet objet rouille, en plus d'altérer sa couleur, la corrosion dégradera petit à petit sa surface. Par opposition à la modification d'apparence, l'ajout de détails géométriques correspond à l'ajout de géométrie supplémentaire, sans modifier l'objet lui-même. Ce sera par exemple le cas pour un objet qui se couvre de neige ou de feuilles mortes.

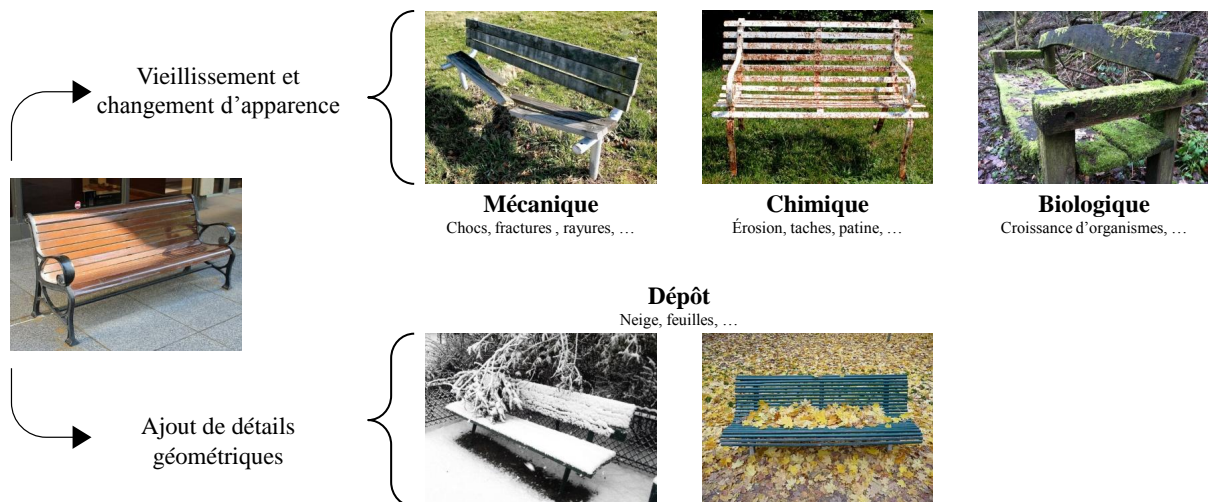


FIGURE 1.4 – Différences entre les méthodes de changement d'apparence et les méthodes d'ajouts de détails géométriques observées au niveau d'un banc.

La prochaine section de ce chapitre détaillera brièvement différentes familles de méthodes de génération procédurale de contenu. Nous examinerons ensuite les techniques de vieillissement et de changement d'apparence, avant d'analyser en détail les méthodes d'ajouts de décorations

géométriques.

2 Génération procédurale de contenu

Dans cette section, nous présentons succinctement les principales méthodes de génération de contenu pour la modélisation de mondes virtuels. Cette section couvre cinq types de contenu : terrains, végétation, routes, villes, et bâtiments. Pour un aperçu plus complet des techniques de modélisation de contenu, nous renvoyons le lecteur au livre [43] et à l'article de synthèse [123] qui traitent la modélisation procédurale dans son ensemble.

2.1 Terrains

Il existe trois familles de création de terrains : simulation, édition, et génération procédurale. La simulation consiste à utiliser des modèles inspirés par la physique, dont la complexité et l'exactitude varient, afin de créer des terrains proches de ceux que l'on peut trouver dans la nature. Les méthodes d'édition sont centrées sur l'utilisateur et pensées pour lui offrir un maximum de contrôle dans le processus de création. Les méthodes de génération procédurale sont très diverses et emploient des outils variés (fractales, bruits, subdivision, etc.) pour générer automatiquement des terrains de surface importante. Les lecteurs intéressés par une analyse plus complète de la génération de terrains se référeront à l'article de synthèse [96] et à la thèse [52].

Simulation de terrains

Le modèle présenté dans [118] applique un processus d'érosion itératif à des terrains dont la composition géologique est définie par un modèle tridimensionnel et la surface par une *carte de hauteur*. Comme le terrain est composé de strates hétérogènes, les étapes d'érosion ont des conséquences différentes en chaque point de la surface du terrain et plus l'érosion est importante, plus elle fait apparaître ces strates (Figure 1.5(a)). La mécanique des fluides, sous la forme des *équations de Navier-Stokes* qui déterminent la pression et la vitesse des fluides, peut déterminer les matériaux érodés et transportés par l'eau [12], permettant de créer des terrains variés allant du cours d'eau calme aux chutes d'eau (Figure 1.5(b)).

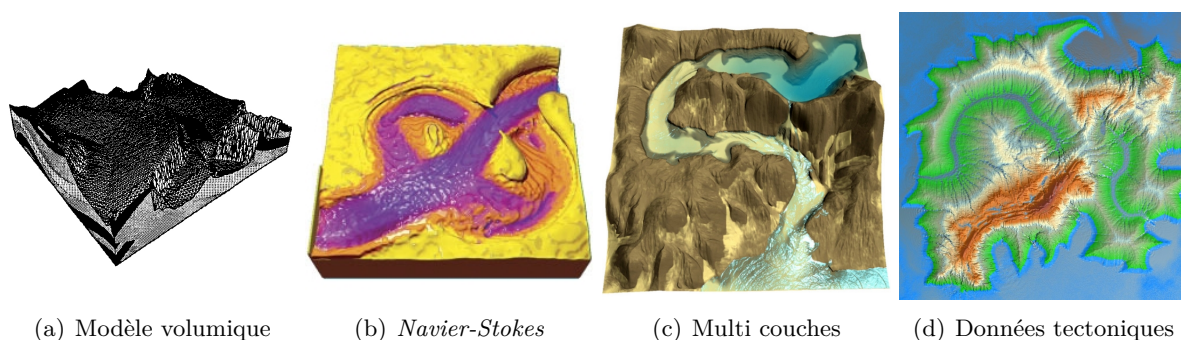


FIGURE 1.5 – Terrains produits par simulation. De gauche à droite, érosion sur un modèle volumique définissant des strates [118], *équations de Navier-Stokes* calculant l'influence de l'eau [12], combinaison de deux types d'érosion sur des couches de matière [136], et utilisation de données tectoniques pour calculer l'élévation et les lignes de séparation des eaux [87].

Un autre modèle physique fonctionnant en temps réel sur processeur graphique utilise deux algorithmes pour déterminer comment l'érosion affecte les différentes couches de matériaux du terrain [136]. Lorsque le flux de l'eau est calme, l'érosion par dissolution prévaut, alors que

l'érosion par force est prépondérante dans le cas des flux d'eau plus importants. Pour pouvoir gérer de grandes surfaces, ce modèle utilise un algorithme de type diviser pour régner qui découpe le terrain en carreaux, calcule indépendamment chaque carreau puis synchronise ensuite leurs bords (Figure 1.5(c)). La méthode présentée dans [87] utilise la composition tectonique du sol, fournie sous forme de cartes vectorielles par l'utilisateur, afin de calculer l'élévation des massifs montagneux et les lignes de partage des eaux. Une fois ces données obtenues, un processus d'érosion hydraulique raffine le terrain créé (Figure 1.5(d)).

Édition de terrains

Bruneton a proposé un modèle vectoriel permettant de peupler de très grands terrains avec des caractéristiques linéaires ou surfaciques (champs, routes, rivières, lacs, etc.) [22]. Cette méthode qui fonctionne sur processeur graphique utilise un *quadtree* et génère à la volée puis stocke plusieurs niveaux de détails pour les données qui doivent être affichées (Figure 1.6(a)). Plusieurs modèles surfaciques d'édition proposent à l'utilisateur de tracer des courbes pour définir les caractéristiques du terrain. Dans celui de Gain [49], l'utilisateur trace une silhouette avant que le terrain soit déformé dans un domaine correspondant à une enveloppe de cette silhouette. Le modèle de Hnadi [62] exploite une méthode de diffusion basée sur des équations différentielles partielles depuis des courbes paramétrées associées à différents types de primitives, tels que des lits de rivière, des falaises ou des fractures (Figure 1.6(b)). Le modèle de Tasse [130] place l'utilisateur dans une vue à la première personne dans laquelle il trace, y compris depuis plusieurs points de vue, les silhouettes complexes qui vont guider la création de crêtes (Figure 1.6(c)).

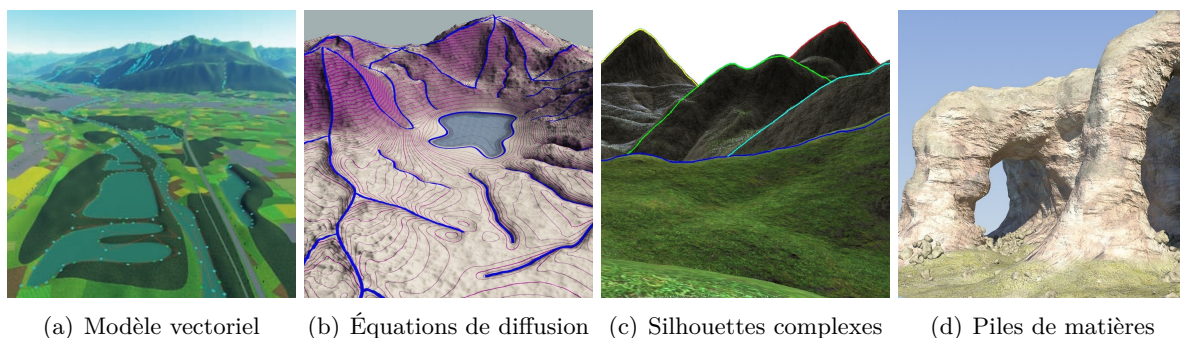


FIGURE 1.6 – Terrains produits par des méthodes d'édition. De gauche à droite : combinaison d'un modèle vectoriel et d'un *quadtree* [22], équations de diffusion [62], silhouettes complexes tracées depuis différents points de vue à la première personne [130] et piles de matières [108].

Une autre approche consiste à utiliser des brosses procédurales interactives évaluées sur processeur graphique pour calculer les élévations du terrain représenté sous forme de carte de hauteur [32]. Contrairement aux modèles d'édition précédents, celui proposé par Peytavie [108] est volumique. Les informations du terrain sont codées sous formes de piles de matières avant d'être transformées en une surface par convolution. Ce modèle permet la création de terrains tridimensionnels contenant des caractéristiques de haut niveau telles que des surplombs, des grottes ou des arches (Figure 1.6(d)).

Génération procédurale de terrain

Kelley a été le premier à proposer une méthode de génération de terrains basée sur les réseaux de rivières [67]. Elle construit un réseau de rivières contenant des segments de formes et de tailles différentes à partir d’un faible nombre de paramètres, et représente la surface du terrain comme une surface sous tension pour effectuer les calculs d’érosion (Figure 1.7(a)). La modélisation à partir de rivières a également été étudiée à l’aide d’un modèle qui combine un algorithme diamant-carré étendu avec une courbe fractale récursive définissant le lit des rivières pour générer la surface du terrain [114]. Un modèle fractal contrôlable localement a également été proposé pour générer des terrains [95]. La modification locale de la dimension de la fractale guidant la génération du terrain permet de modifier aisément certaines de ses propriétés, comme l’altitude. Un système d’érosion hydraulique simple crée ensuite les vallées et les cours d’eau (Figure 1.7(b)). Un autre modèle fractal déterminant l’élévation du terrain à partir des rivières a été proposé par Teoh [131]. Il génère d’abord les crêtes puis le réseau hydrographique à partir des embouchures en découpant le terrain en cellules, avant de déterminer l’élévation des cellules depuis les embouchures en remontant le cours des rivières.

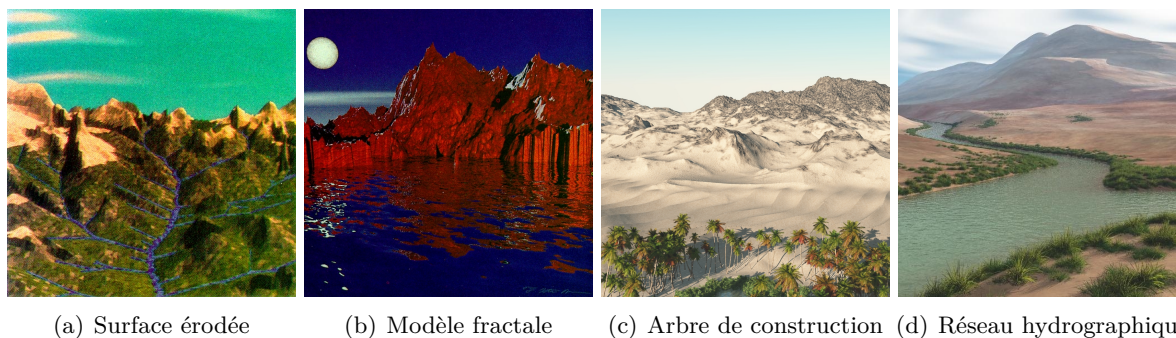


FIGURE 1.7 – Terrains générés procéduralement. De gauche à droite : réseau de rivières érodant une surface sous tension [67], modèle fractale [95], arbre de construction combinant des primitives [54], et réseau hydrographique pilotant la création de l’arbre de construction [53].

Géneveaux a présenté un modèle surfacique de terrains basé sur un arbre de construction combinant des primitives définies par des fonctions lipschitziennes par des opérateurs (mélange, remplacement, homothétie, etc.) [54]. Ce modèle très compact permet de visualiser des terrains à différentes échelles, aussi bien comme un maillage classique que par lancer de rayon (Figure 1.7(c)). L’auteur a utilisé ce modèle pour créer des terrains à partir de réseaux hydrographiques construits procéduralement [53]. Deux fonctions contrôlent les pentes des rivières et des vallées et pilotent le processus de génération du réseau hydrographique qui détermine les lignes de crêtes et les zones de partage des eaux en fonction d’une classification des types de cours d’eau. L’arbre de construction du terrain créé comporte des primitives de rivières spécialisées qui font apparaître des caractéristiques de haut niveau tel que des bancs de sable et des berges détaillées (Figure 1.7(d)).

2.2 Végétation

La modélisation de végétation a fait l’objet de beaucoup d’attention de la part de la communauté scientifique, notamment suite aux travaux de Lindemayer qui a inventé et publié les *L-System* dans deux articles successifs [74] [75]. Prusinkiewicz a été le premier à utiliser les *L-System* pour modéliser des végétaux [113], et ce formalisme s’est avéré être un outil très puissant et extrêmement bien adapté à la génération de végétation. Une version étendue, nommée open

L-System, a été conçue afin de créer des végétaux plus réalistes en considérant l'environnement lors de leur développement [80].

Plusieurs modèles de génération ont été développés dans l'objectif de contrôler précisément l'apparence de la végétation virtuelle. L'approche publiée dans [115] consiste à utiliser des *L-system* paramétrés par des informations exogènes pour élaguer la structure des végétaux (Figure 1.8(a)). Un arbre peut également être considéré comme une structure qui s'auto-organise au cours de son développement [102]. L'exposition à la lumière, l'environnement et différentes fonctions de pondération sont alors utilisés pour guider la croissance de la structure de l'arbre par colonisation de l'espace (Figure 1.8(b)).

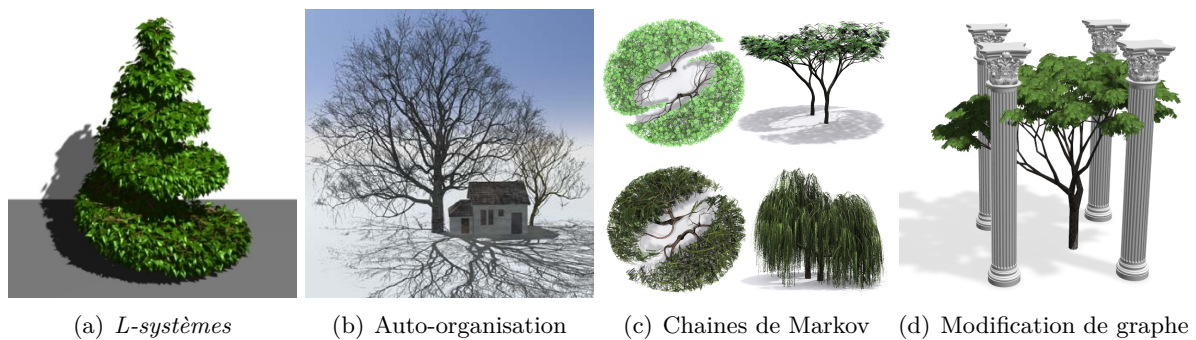


FIGURE 1.8 – Arbres procéduraux. De gauche à droite : élagage au moyen de *L-system* pilotés par des informations exogènes [115], structure auto-organisée [102], Monte-Carlo par chaînes de Markov [129] et modification de graphe [111].

Talton a présenté une méthode de contrôle de grammaires procédurales exploitant les méthodes de Monte-Carlo par chaînes de Markov, adaptée entre autres à la modélisation de la flore [129]. Elle produit des modèles qui ont à la fois une apparence naturelle et qui respectent avec précision des paramètres fournis en entrée (Figure 1.8(c)). La méthode interactive présentée dans [111] n'est pas destinée à créer des arbres, mais à adapter des arbres existants en fonction de l'ensoleillement et l'espace disponible, tout en les altérant aussi peu que possible. Les branches peuvent être pliées ou élaguées, et le feuillage est recalculé après les transformations qui interviennent dans le graphe définissant la structure de l'arbre (Figure 1.8(d)).

Au vu de la très grande quantité de travaux traitant de la modélisation de végétaux, nous avons simplement survolé ce domaine. De plus amples informations sont disponibles dans l'article de synthèse [18] et dans le livre [39]. Nous détaillons les méthodes de génération de végétaux liées aux modifications d'apparence (herbes et feuilles) dans la section 4.2 de ce chapitre.

2.3 Routes et réseaux routiers

Par opposition à la génération de villes qui consiste entre autres à générer des ruelles et des rues dont la longueur et la largeur sont faibles, la génération de routes a pour but la création de voies beaucoup plus longues, plus larges, et potentiellement plus tortueuses. McCrae a proposé un système d'édition de réseau en trois dimensions et a démontré son fonctionnement en l'utilisant pour créer des routes [79]. Les esquisses créées par les utilisateurs sont analysées par le système et converties en un ensemble de clothoïdes, des courbes dont la courbure varie linéairement en fonction de la longueur d'arc, et le système génère automatiquement des ponts et des tunnels en respectant des contraintes de courbure (Figure 1.9(a)). Le modèle d'édition architecturale présenté par Cabral [24] est centré sur la création de grandes structures complexes, y compris

des routes. La modélisation se fait par morceaux : le système combine différents morceaux de route qui sont adaptés pour se joindre correctement, tout en préservant leurs caractéristiques telles que les angles ou l'orientation des murets (Figure 1.9(b)).

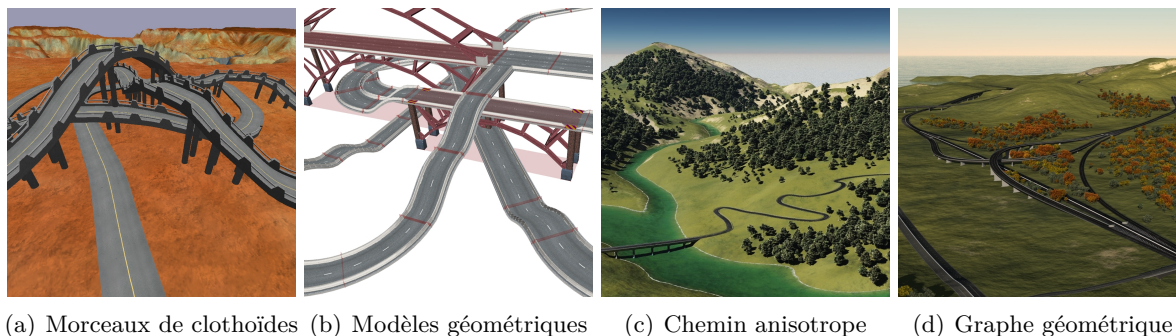


FIGURE 1.9 – Routes procédurales. De gauche à droite : assemblage de morceaux de clothoïdes [79], déformation et combinaison de modèles géométriques [24], algorithme de plus court chemin anisotrope [51] et réseau hiérarchique basé sur un graphe géométrique [50].

La méthode présentée par Galin dans [51] utilise un algorithme de plus court chemin anisotrope pondéré pour créer la route qui relie deux points d'un terrain. La trajectoire de la route minimise une fonction de coût qui prend en compte différents paramètres comme la pente du terrain et la présence d'obstacles naturels tels que les lacs, les rivières et les forêts. Le modèle géométrique de la route est créé en terrassant le terrain le long de la trajectoire et en utilisant des objets paramétrés comme des ponts pour franchir des zones spécifiques (Figure 1.9(c)). Contrairement à cette méthode qui est centrée sur la génération de la géométrie de la route, la méthode proposée par Galin dans [50] se focalise sur la création des trajectoires d'un réseau routier hiérarchique à grande échelle, reliant différents types de routes connectant des dizaines de villes et de villages. Elle combine un algorithme de génération de graphe géométrique basé sur une distance non euclidienne avec un algorithme de fusion de trajectoires qui crée des jonctions entre les différents types de routes dont les modèles géométriques sont générés avec la méthode précédente (Figure 1.9(d)).

2.4 Villes

Les méthodes de génération de ville se découpent en trois étapes (Figure 1.10). La première est la création d'un réseau routier complexe composé de rues et de routes. La seconde étape consiste à déterminer quelles sont les parcelles de terrains qui recevront des bâtiments, et selon la complexité de la méthode, le type de bâtiment qui leur est associé (industriel, commercial, résidentiel, etc.). La dernière étape détermine l'empreinte des bâtiments et leur distribution dans les parcelles. Il existe de rares méthodes de génération de villes qui génèrent également des bâtiments, mais les modèles créés dans ces cas sont trop simples pour être convaincants.

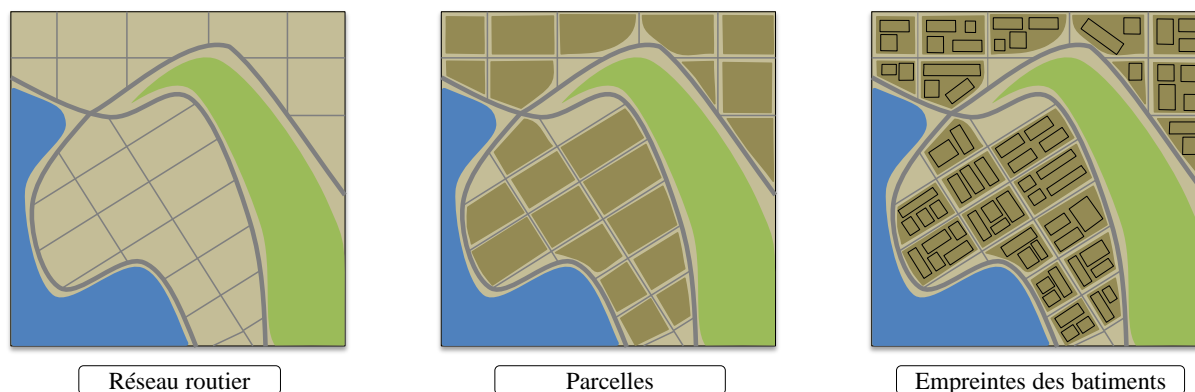


FIGURE 1.10 – Les villes procédurales sont générées en trois étapes : le découpage du réseau routier crée des parcelles sur lesquelles sont distribués les bâtiments.

La première méthode de génération de ville a été publiée par Parish et Müller [104]. À partir de cartes codant différentes informations (eau, élévation du terrain, population), un réseau routier est généré en utilisant un *L-System* modifié. Le réseau routier obtenu divise l'espace au sol en lots, de petites zones contenant les bâtiments générés par un autre *L-System* paramétrique et stochastique (Figure 1.11(a)). Esch a proposé une méthode interactive exploitant des champs de tenseurs utilisés pour créer des graphes guidant la génération des routes [44]. Il s'agit de la première méthode de génération de ville qui combine une approche procédurale avec de l'édition. Elle offre un haut niveau de contrôle à l'utilisateur qui peut directement modifier le graphe obtenu pour influencer le résultat final (Figure 1.11(b)). La méthode présentée par Aliaga repose sur l'analyse d'images aériennes du monde réel [2]. Le réseau routier et les parcelles de terrain qui contiennent des bâtiments sont extraites des images ; et la reconstruction est ensuite réalisée par une approche basée exemple. Cette méthode peut étendre fidèlement une parcelle de ville existante en créant de nouvelles données, ou connecter deux parcelles topologiquement très différentes (Figure 1.11(c)).

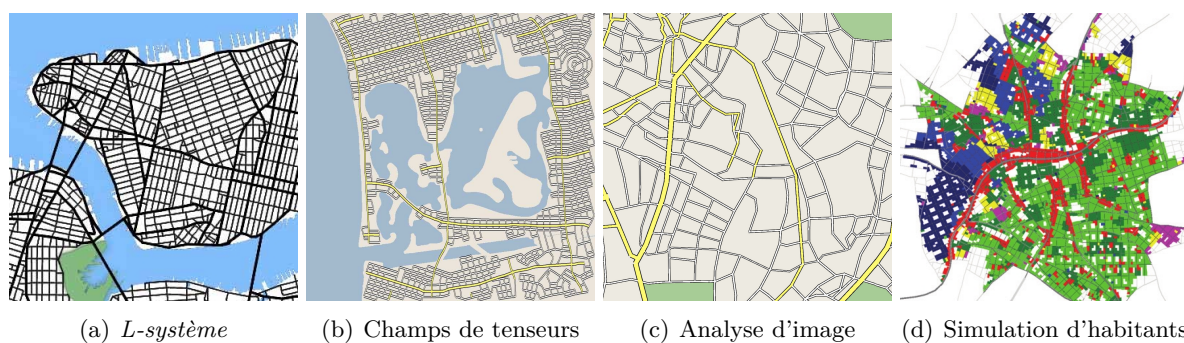


FIGURE 1.11 – Réseaux de rues et de routes produits par des méthodes de génération procédurale de ville. De gauche à droite, les outils employés sont les *L-system* [104], les champs de tenseurs [44], l'analyse d'image [2], et la simulation du comportement des habitants virtuels [139].

Le modèle présenté par Vanegas utilise les informations du relief et les comportements des habitants fictifs, qui sont influencés par différents paramètres (population, prix du terrain, travail, etc.), pour piloter la simulation qui guide le développement de la ville [133]. La ville générée reflète ces contraintes : la taille des routes est par exemple dépendante de la quantité de véhicules qui y circulent. Un autre modèle de simulation modélisant l'évolution d'une ville en fonction du temps a été proposé par Weber [139]. La simulation exploite des informations comme la classi-

fication du terrain (zones résidentielles, commerciales, industrielles), son prix ou le nombre de nouvelles rues créées par année pour guider la génération de la géométrie de la ville. Le résultat créé à chaque pas de la simulation n'est pas définitif, ce qui permet de déplacer des portions de la ville tel que les zones industrielles au fils des ans en fonction des résultats de la simulation (Figure 1.11(d)).

2.5 Bâtiments

La génération de bâtiments fonctionne généralement en trois étapes. Il est d'abord nécessaire de construire l'empreinte au sol d'un bâtiment pour ensuite générer son enveloppe extérieure, avant de finalement créer sa géométrie : détails des façades, des toits, etc. (Figure 1.12).

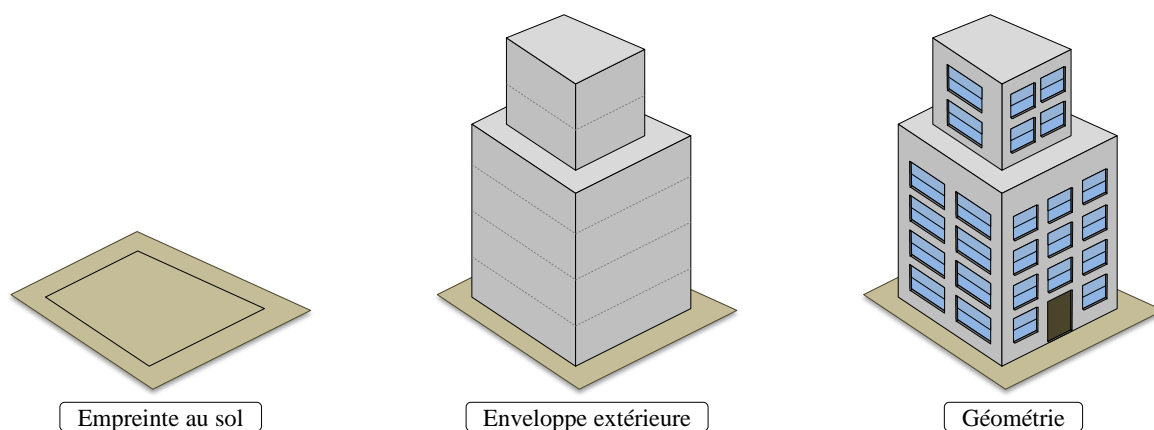


FIGURE 1.12 – Les trois étapes de la génération procédurales de bâtiment.

Les premiers bâtiments générés automatiquement sont apparus dans [104]. La géométrie des gratte-ciels était simple, et leurs façades étaient représentées exclusivement par des textures (Figure 1.13(a)). Wonka a introduit les *Split-Grammars* deux ans plus tard [144]. Cette grammaire est basée sur un ensemble de formes géométriques qu'elle découpe et subdivise récursivement en fonction de règles. Les bâtiments produits sont beaucoup plus détaillés que dans l'approche précédente et les caractéristiques des façades (portes, fenêtres, rebords, balcons, etc.) sont modélisées géométriquement (Figure 1.13(b)). Un nouveau type de grammaire nommé *CGA Shape* a ensuite été proposé par Müller [91]. C'est une extension en trois dimensions des grammaires de forme de Stiny [124] et des *Split Grammars*. Des règles supplémentaires permettent des transformations géométriques (rotation, translation, homothétie) et rendent cette grammaire plus versatile. Elle gère également les occlusions entre les formes, ce qui permet de générer des façades dont les ouvertures sont cohérentes (Figure 1.13(c)).

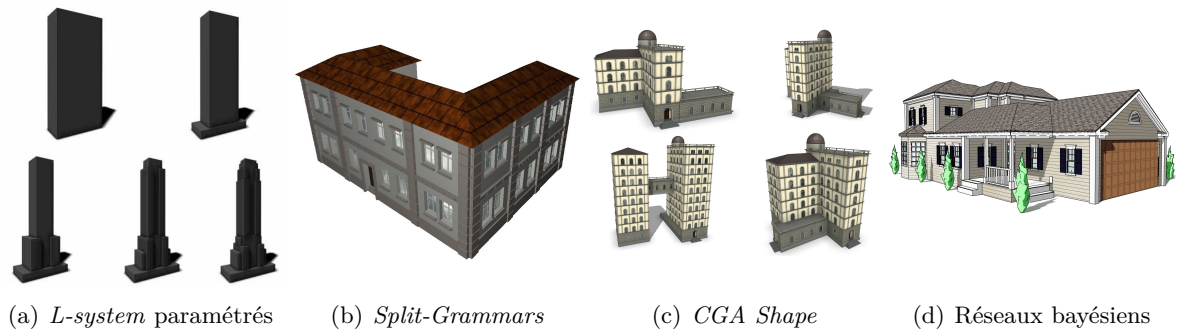


FIGURE 1.13 – Bâtiments procéduraux. De gauche à droite, les méthodes de génération utilisent des *L-systeme* stochastiques paramétrés [104], des grammaires de formes *Split-grammars* [144], des grammaires de formes *CGA-Shape*, et des réseaux bayésiens [86]

Müller a proposé d'utiliser des images de façades existantes pour en générer de nouvelles [92]. Cette méthode détecte les éléments répétés dans les façades afin d'en créer une représentation au moyen d'un arbre de forme. Un modèle polygonal est ensuite construit à partir de ces informations, et une version corrigée pour supprimer les artefacts de la texture fournie en entrée est finalement plaquée sur le modèle créé. Récemment, Mereel a proposé une approche utilisant des réseaux bayésiens exploitant des données réelles et conçue pour générer des habitations résidentielles à un ou deux étages [86]. Contrairement aux méthodes classiques, elle ne se base pas sur une empreinte au sol : elle itère sur les contraintes exprimées par un réseau bayésien pour générer un plan en deux dimensions du bâtiment, pour ensuite générer sa géométrie, y compris les murs intérieurs qui séparent les pièces. L'exploitation de contraintes issues du monde réel permet la génération de modèles géométriques très détaillés et très fidèles à la réalité terrain (Figure 1.13(d)).

2.6 Synthèse

Il existe une très grande quantité de méthodes extrêmement diversifiées pour générer tout type de contenu graphique nécessaire à la création de mondes virtuels. Certains domaines, comme la génération de terrain ou de végétation, ont été particulièrement étudiés dès les débuts de l'informatique graphique. Cependant, mêmes dans ces domaines, les méthodes existantes ne suffisent pas à répondre à des besoins croissants. Le principal verrou scientifique actuel de la génération de contenu graphique est la gestion simultanée de différentes échelles. Il existe des méthodes spécialisées dans la génération de contenu extrêmement détaillé (plantes, bâtiments, etc.) ou à l'inverse dans la génération de contenu extrêmement vaste (réseaux routiers, terrain) ; mais il est extrêmement difficile de créer du contenu partageant ces deux caractéristiques au sein d'un même système.

3 Vieillessement et changements d'apparence

Les méthodes qui modifient l'apparence d'un objet sont regroupées sous la classification anglophone *aging and weathering*, que l'on traduit généralement comme *vieillessement et changements d'apparence*. Dans la grande majorité des cas, ces méthodes visent à reproduire un phénomène naturel le plus fidèlement possible. Elles s'appuient donc majoritairement sur des modèles physiques, climatologiques ou biologiques pour représenter les interactions qui ont lieu dans le monde réel, avec différents niveaux d'approximation. Ce domaine qui regroupe un nombre très important de phénomènes est très étudié en informatique graphique, puisque sans ces phénomènes, les scènes virtuelles sont artificielles et difficilement crédibles. Ces méthodes sont particu-

lièrement utilisées en synthèse d'image réaliste dans le contexte des effets spéciaux de l'industrie cinématographique.

Le changement d'apparence d'un objet peut être provoqué par des causes très variées : conditions climatiques, température, qu'elle soit élevée ou faible, exposition à des substances chimiques, modification de sa composition biologique, contacts, frottements ou collisions avec d'autres objets, etc. Ces phénomènes se manifestent par l'apparition de patine et d'oxydation sur les métaux, par la colonisation de surfaces par des structures biologiques telles que les moisissures, les mousses ou les lichens, par un dépôt de pollution à la surface des bâtiments et des monuments, ou par la formation de rides qui déforment les objets organiques. Ces deux énumérations ne sont bien entendu pas exhaustives. Pour un aperçu complet portant sur le vieillissement et les changements d'apparence, nous renvoyons le lecteur à l'article de synthèse [89] et au livre [42].

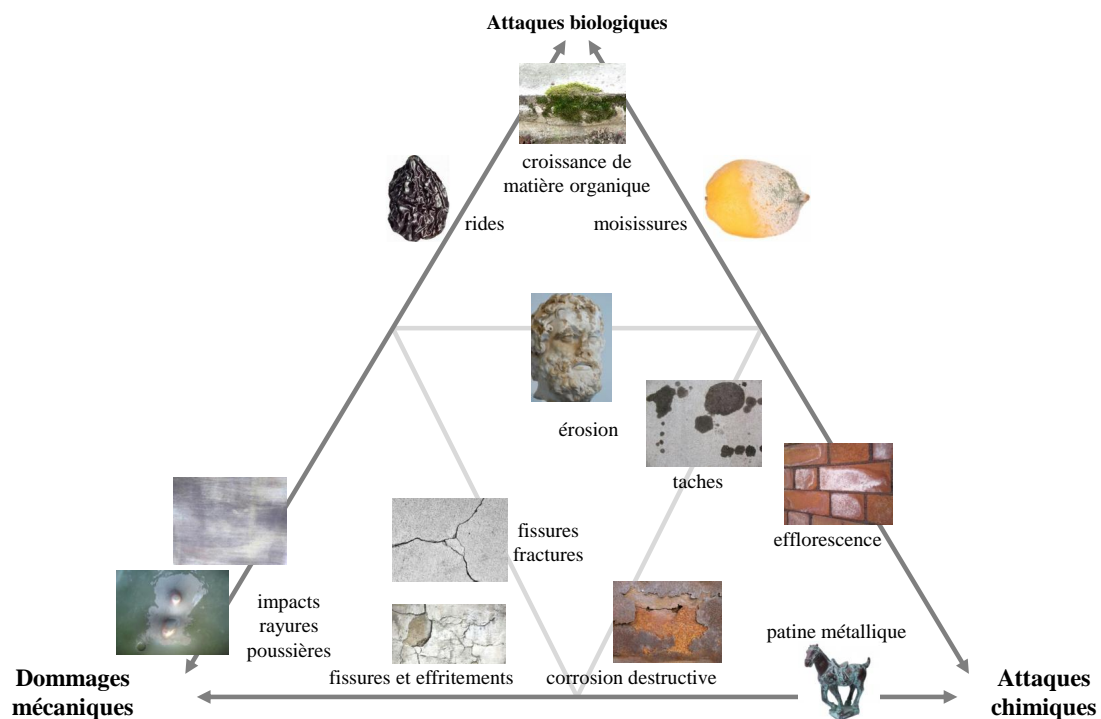


FIGURE 1.14 – Répartition des méthodes de vieillissement et de changements d'apparence selon leur nature d'après [89].

Certaines méthodes sont générales ou ne traitent pas d'un phénomène en particulier. Elles ont introduit des concepts généraux comme les textures de vieillissement ou le *bump mapping* pour produire une grande variété d'effets. Dès 1978, Blinn proposait un moyen pour introduire des détails sur les surfaces afin de casser leur apparence artificiellement lisse [13]. Becket et Badler ont été parmi les premiers à proposer une méthode faite pour rompre l'homogénéité des surfaces et faire apparaître des imperfections, notamment de la rouille, en synthétisant des textures à l'aide de fractales de différentes dimensions [7]. La méthode des *gamma-ton*, introduite dans [28], repose sur le transport de différentes informations par des particules rebondissant à la surface des objets. Très versatile, elle peut modéliser des phénomènes très différents, tels que la rouille, la croissance de mousse ou l'érosion.

Ces techniques générales mises à part, les méthodes de vieillissement et de changements d'ap-

parence peuvent être classées en trois catégories : dommages mécaniques, attaques biologiques, et attaques chimiques (Figure 1.14).

3.1 Dommages mécaniques

Cette catégorie est la plus vaste de cette classification puisque qu'elle regroupe des effets très variés. Les dommages mécaniques peuvent avoir des origines très diverses : les frottements engendrent des rayures ou érodent les matériaux, les impacts donnent lieu à des creux, des bosses, des fractures et des fissures, etc.

3.1.1 Fractures



FIGURE 1.15 – Champs de contrainte [64].

Différentes approches ont été proposées pour générer des fractures. Dans [34], les fractures sont créées en éditant des courbes de profil à la surface des objets, puis en creusant leur surface en fonction des matériaux qui les composent. La modélisation de fractures par une approche à partir d'exemples a été étudiée dans [55]. Elle se base sur une collection de photographies de fracture. Récemment, Chen a proposé un modèle de fracture fonctionnant en deux étapes [29]. Ce modèle génère d'abord une version grossière des fractures en s'appuyant sur la résistance des matériaux, avant de raffiner les objets à la frontière des fractures. La

méthode procédurale de modélisation de fracture présentée dans [78] gère les matériaux solides tels que le verre ou la pierre et met l'accent sur le contrôle. Dans [64], les fractures sont modélisées à partir d'un champ de contraintes paramétrable (Figure 1.15). Ce modèle permet de créer des fractures dans des matériaux très hétérogènes, tels que de la pierre, du verre, de la boue ou encore du bois. La méthode présentée dans [61] génère des textures de fractures à partir d'un modèle physique basé sur un système masse-ressort. O'Brien a présenté un modèle de fracture en trois dimensions dédié aux matériaux solides [99]. Ce modèle a ensuite été étendu afin de gérer également les matériaux ductiles [98]. Nous renvoyons le lecteur à l'article de synthèse [90] pour une analyse approfondie de la modélisation de fractures.

3.1.2 Impacts, chocs et rayures

Paquette [103] est le premier à s'être intéressé aux conséquences des impacts et des chocs sur le vieillissement (Figure 1.16). Ses travaux modélisent les imperfections et les détériorations dues aux coups portés par un objet sur la surface d'un autre en raffinant la surface de l'objet cible avec un maillage adaptatif. Mérillou *et alii* ont travaillé sur le rendu réaliste de rayures et d'éraflures. Leur modèle physique présenté dans [85] utilise des mesures réelles pour calculer le rendu de rayures et d'éraflures en combinant textures et BRDF. Les rayures et les éraflures ont également été le sujet des travaux de Bosch dans [17]. À partir de données telles que la forme de l'outil utilisé, la force appliquée sur l'outil ou les propriétés du matériau dont la surface est rayée, la micro-géométrie des rayures est générée puis rendue avec une BRDF.



FIGURE 1.16 – Maillage adaptatif [103]

3.1.3 Transport de matière

Dorsey [41] a proposé une des méthodes fondatrices de ce domaine : la modélisation des précipitations que subissent les objets situés en extérieur, qui nettoient les parties les plus ac-

cessibles des objets, et au contraire, salissent les zones les moins accessibles. Mérillou *et alii* ont proposé un système simulant un phénomène proche, l'accumulation de la pollution sur les bâtiments, dans [83]. La pluie qui lave les zones accessibles y empêche la formation de gypse. Au contraire, dans les zones peu accessibles par la pluie, le minerai de gypse responsable de la coloration noire des surfaces polluées peut se cristalliser.

3.1.4 Poussières

Blinn a été le premier à proposer une méthode de rendu de poussière [14]. Cette approche utilise un ensemble de particules, trop petites pour que leur influence individuelle soit visible, pour créer la couche de poussière. Un autre modèle de poussière a été proposé par Hsu [63]. Il est dérivé du modèle d'éclairage de Phong [110] : les sources de poussière remplacent les sources de lumière, et la quantité de poussière qui se dépose en un point est proportionnelle au cosinus entre la source de poussière et la normale de la surface. Pour améliorer les résultats, la quantité de poussière est modulée par l'accessibilité. Cependant, ces deux modèles traitent la modélisation de la poussière comme une problématique de rendu, ce qui est très simplifié puisqu'ils ne tiennent pas compte du caractère volumique de la poussière. Adachi a proposé un nouveau modèle volumique de modélisation de poussière [1] (Figure 1.17). Ce modèle utilise les *Shell Texture Fonctions* issues de [27] et permet de modéliser différents types de poussière, comme la poussière fibreuse issue du tissu ou les poussières diffuses, beaucoup plus fines. Wong a proposé une méthode hybride, à mi-chemin entre la simulation physique et l'édition manuelle de texture, capable de représenter de la poussière, de la patine ou la détérioration de la peinture [143].



FIGURE 1.17 – Poussière volumique [1]

3.2 Attaques biologiques

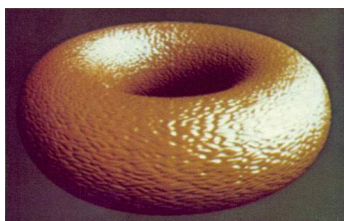


FIGURE 1.18 – Perturbation de la surface [13]

L'une des caractéristiques les plus visibles des attaques biologiques est l'apparition de rides à la surface des objets qui vieillissent. La première façon de représenter des rides a été proposée par Blinn [13]. Cette technique consiste à utiliser une fonction pour perturber les sommets de la surface d'un objet afin de casser son homogénéité. Cette technique purement géométrique est très facile à contrôler et peut être utilisée pour reproduire une très grande variété de caractéristiques au delà des rides, comme les irrégularités de la surface des fruits ou des légumes, le gondolement de la peinture, etc. Elle est très largement utilisée et connue sous le nom de *bump-mapping* (Figure 1.18).

Les rides ont principalement été étudiées dans le contexte humain, étant donné leur importance dans la représentation réaliste des visages et des animations faciales. Le modèle proposé dans [145] comporte trois couches : la peau sous forme de maillage triangulaire, les tissus conjonctifs, et les muscles sous la forme de carreau de B-splines. La couche des tissus conjonctifs connecte la peau aux muscles avec un système masse-ressorts. Les déformations à la surface de la peau sont modélisées de deux façons différentes : avec un modèle plastique pour représenter les rides permanentes dues à la vieillesse, et un modèle élastique pour les rides temporaires dues aux expressions faciales. Le principal problème de ce modèle est que la représentation de la peau est

surfactive, ce qui implique qu'elle n'a pas de réelle épaisseur.

Une amélioration de ce modèle afin de considérer la peau comme une structure comportant plusieurs couches a ensuite été proposée [16]. Chaque couche de la structure correspond à un des tissus spécifiques de la peau, qui ont des propriétés biomécaniques différentes. Les rides permanentes et temporaires sont cette fois modélisées avec une technique basée sur les éléments finis. Le système combine également des cartes de rides issues de photos réelles pour créer la texture appliquée au modèle géométrique.

Sumner a étudié la formation de motif lors de la croissance du lichen dans sa thèse [126]. Ses travaux reposent sur le modèle d'instabilité de Saffman-Taylor et les équations de croissance laplacienne associée. Desbenoit a présenté un système de modélisation qui simule la propagation et la croissance des lichens [35]. La propagation des lichens est cette fois réalisée par un système de diffusion-agrégation limité ouvert, qui est influencé par l'environnement. Ce système guide ensuite l'instanciation de modèles géométriques de lichens (Figure 1.19).



FIGURE 1.19 – Croissance de lichens [35]

3.3 Attaques chimiques

Dans [41], Dorsey et Hanrahan ont modélisé la patine du cuivre en considérant les surfaces métalliques comme une succession de strates combinées à l'aide d'opérateurs de recouvrement, d'érosion et de polissage. Chang et Shih ont proposé une méthode dédiée à un autre type de patine - celle qui se forme sur les bronzes enterrés - dans [25]. Contrairement à la patine qui n'affecte que la couleur d'un objet, la corrosion peut également détruire une ou des parties de l'objet puisque ce mécanisme chimique peut lui faire perdre de grandes quantités de matière. Un modèle gérant différentes formes de corrosion a été proposé par Mérillou [84]. Il permet de reproduire la corrosion uniforme (attaque similaire sur toute la surface), la corrosion par piqure (attaque très localisée), la corrosion galvanique (à l'interface de deux métaux) et la corrosion de crevasse (due à des différences d'oxygénation) (Figure 1.20).

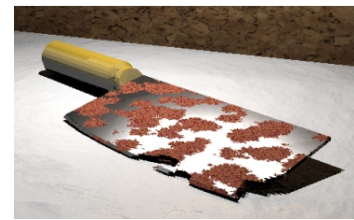


FIGURE 1.20 – Modélisation de rouille [84]

Mérillou a étudié comment la présence de sel dans les matériaux de construction affecte leur décomposition et participe au vieillissement des bâtiments [82]. La porosité, la concentration en sel des matériaux, l'illumination et l'humidité sont prises en compte dans ce modèle. Chang [26] a présenté un modèle de synthèse de texture basé sur un *L-system* pour modéliser l'apparence rouillée d'objets métalliques qui se corrodent au contact de l'eau de mer. Dorsey a proposé un modèle de vieillissement de pierre [40] qui simule les flux qui se déplacent à la surface, ainsi que la dissolution puis la recristallisation des minéraux sous la surface poreuse de la pierre.

3.4 Synthèse

Le domaine du vieillissement et des changements d'apparence est très vaste, et il regroupe une très grande quantité de méthodes hétérogènes qui font appel à des principes et des structures de données extrêmement variés. La principale faiblesse de ces méthodes est leur manque de versatilité. Comme elles sont destinées à simuler la réalité, l'amélioration de la qualité des

résultats qu’elles sont capables de produire se fait au prix de l’augmentation de leur complexité et de leur spécialisation, ce qui rend leur exploitation compliquée dans un contexte de production industrielle. Seules quelques méthodes [13] [7] [28] sont capables de reproduire différents phénomènes.

4 Ajouts de géométrie

Les méthodes d’ajout de géométrie sont très variées. Il existe des méthodes pour générer de la végétation (feuille, herbe, lichen, mousse, etc.), une couche de neige ou de glace, le dépôt de la poussière, etc. Les techniques utilisées pour les modéliser sont aussi hétérogènes que les phénomènes concernés.

4.1 Phénomènes hivernaux

Nous regroupons dans les phénomènes hivernaux la neige et la glace, phénomènes qui ont été largement étudiés. De très nombreuses méthodes traitant des différentes caractéristiques de la neige et de la glace existent dans l’ensemble de la littérature scientifique, et dans le domaine de la physique en particulier : que ce soit pour la modélisation d’avalanches [132], le calcul des propriétés optiques de la neige [15] ou l’analyse de l’albédo terrestre [138] [142], ces phénomènes ont reçu énormément d’attention de la communauté scientifique depuis un grand nombre d’années. Cependant, dans un souci de pertinence et de concision, nous nous intéresserons dans cet état de l’art exclusivement aux méthodes liées à la synthèse d’images.

4.1.1 Génération de neige

Nous regroupons les méthodes de génération de neige selon trois catégories : les systèmes à base de particules, les techniques de déplacement de surface, et les méthodes utilisant des piles de matières.

Génération de neige à base de particules

Les méthodes à base de particules créent la couche de neige à partir de ses flocons (Figure 1.21). Selon leur granularité, elles représentent chaque flocon ou chaque amas de flocons par une particule. Ces méthodes fonctionnent en deux étapes : simulation du déplacement des particules puis reconstruction de la surface de neige. La boucle de simulation comprend deux phases. À chaque itération de la première, la nouvelle position des particules est calculée en fonction de leur déplacement (qui peut prendre en compte différents éléments perturbateurs, tels que le vent ou d’autres flux d’air). Lors des itérations de la seconde phase, les particules qui sont arrivées au contact de la surface d’un objet ou d’autres particules déjà stabilisées se déposent et s’agrègent. Lors de la seconde étape, la surface de neige est construite à partir de l’ensemble des particules qui se sont déposées sur l’objet.

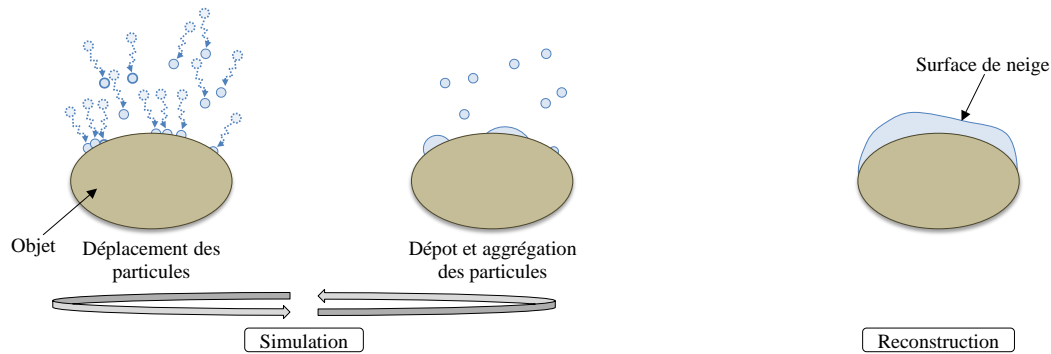


FIGURE 1.21 – Les deux étapes de la génération de neige à base de particules : simulation et reconstruction.

Nishita a été le premier à proposer une méthode de modélisation et de rendu de neige en 1997 [97]. La modélisation de la neige est calculée à partir d'une distribution de metaboules qui sont des primitives ponctuelles de surfaces implicites caractérisées par un centre, un rayon et une densité au centre (la densité correspondant à la quantité de neige contenue par la metaboule). Le rendu de la surface de neige, qui est l'isosurface issue de l'union des metaboules distribuées dans la scène, est effectué en prenant en compte la diffusion de la lumière à l'intérieur de la neige, pour tenir compte du caractère volumique de ce matériau (Figure 1.22).



FIGURE 1.22 – Surface implicite [97]



FIGURE 1.23 – Déplacement de flocons [45]

La méthode proposée par Fearing [45] fonctionne « à l'envers » : au lieu de lancer des particules depuis l'atmosphère et de calculer leur trajectoire descendante puis de déterminer leur position sur le sol, elle tire des particules depuis le sol vers l'atmosphère, avec une trajectoire perturbée. Les particules dont la trajectoire est bloquée par des éléments de la scène ne peuvent atteindre le ciel, et ne contribuent donc pas à la quantité de neige qui se dépose au sol, ce qui gère l'accessibilité de façon très naturelle. Cette démarche permet ainsi à chaque surface recevant de la neige de spécifier la quantité de particules qui lui est associée, ce qui détermine la précision de sa surface de neige ainsi que le temps de calcul correspondant. Une fois la quantité de neige déterminée depuis les particules, une série de micro-avalanches est utilisée pour stabiliser et lisser la couche de neige. Ce modèle permet d'obtenir de bons résultats, y compris sur de petits éléments, comme les branches d'arbre (Figure 1.23). Cependant, le coût des calculs augmente avec la complexité de la scène, et il n'est pas envisageable d'utiliser ce modèle sur de grandes scènes comprenant beaucoup de détails.

Muraoka [94] calcule le déplacement des flocons dans un flux d'air en trois dimensions. Cette méthode est également basée sur la propagation de la chaleur et prend en compte la chaleur du soleil, du sol, et la chaleur radiante émise par les différents objets. Le flux d'air qui transporte les flocons est géré par un ensemble de champs de vortex et déplace les flocons en prenant en compte les mouvements de l'air, la gravité, ainsi que des perturbations stochastiques ; jusqu'à ce que les flocons atteignent la surface d'un objet. Les flocons sont ensuite déplacés sur la surface

de l'objet en calculant les forces de mouvement, jusqu'à ce qu'un équilibre d'adhérence à la surface soit atteint. Bien qu'elle présente l'avantage de gérer la fonte de la neige et d'obtenir des résultats précis, cette méthode est coûteuse et ne permet de simuler une couche de neige que sur de très petites scènes.

La méthode de Fearing a été améliorée par Feldman [46] qui l'a combinée aux équations de Navier-Stokes et un schéma d'intégration semi-Lagrangien, des outils utilisés en mécanique des fluides. Ces calculs permettent de modéliser le flux qui se déplace dans une scène, et de suivre ensuite avec beaucoup de précision les déplacements des flocons de neige. La cohérence de la couche de neige est assurée par une série de calculs qui lisse en chaque point d'accumulation la hauteur de neige en fonction des points d'accumulations voisins, et assure que la pente de la surface ne dépasse jamais un seuil critique. Une fois cette étape de stabilisation effectuée, une carte de hauteur est extraite depuis les points d'accumulation, et affichée en tant que surface de subdivision en utilisant l'algorithme de Catmull-Clark (Figure 1.24). Cette méthode produit des surfaces de neige remarquablement plus complexes que [45], permettant de modéliser des phénomènes comme la formation de congères.

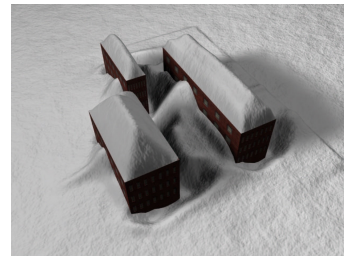


FIGURE 1.24 – Simulation de chute de neige [46]

Wang [137] a utilisé un autre outil mathématique, les équations de Boltzmann, pour améliorer le calcul du déplacement des flocons de neige. Cet outil permet de créer un modèle en trois dimensions du vent et de simuler le déplacement, le dépôt et l'érosion des particules de neige en prenant en compte la gravité et les collisions entre flocons. Le système est interactif, puisque la force et la direction du vent peuvent être modifiées au cours de la simulation, tout en maintenant un nombre d'images par seconde suffisant pour un affichage en temps réel. Cette méthode permet de représenter des caractéristiques évoluées de la neige, comme les congères ou l'accumulation contre les surfaces verticales.

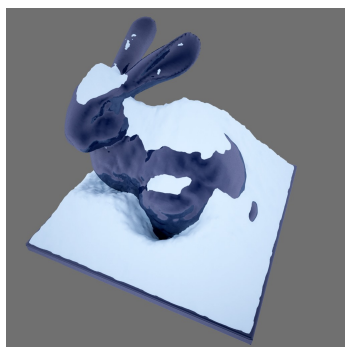


FIGURE 1.25 – Agrégation de particules [60]

très faible. De plus, les temps de calcul sont très élevés : environ trois heures ont été nécessaires pour calculer l'image située à gauche de ce paragraphe, alors que la scène ne mesure qu'un mètre carré.

Dans [60], la modélisation de la couche de neige est faite à l'aide d'une technique inspirée par le comportement physique de la neige. À chaque pas de temps, des contraintes de stabilité granulaire déterminent quelles sont les particules qui peuvent adhérer à l'objet ou à la couche de neige, ou quelles sont les particules déjà agrégées qui peuvent se détacher de la surface sous l'influence de la gravité ou du vent. Un modèle de surface implicite est utilisé pour mettre à jour la surface de la neige à chaque agrégation d'une particule. Comme l'angle de repos maximum des nouvelles particules dépend de la température, les résultats obtenus varient avec elle. Cette méthode produit des résultats de grande qualité. Cependant, elle ne fonctionne que sur des scènes dont la taille est

Génération de neige par déplacement de surface

Les méthodes de génération de neige par déplacement de surface fonctionnent en deux étapes (Figure 1.26). Dans ces méthodes, la couche de neige est générée à partir de la géométrie des

objets sur laquelle la neige se dépose. Un vecteur de déplacement est calculé pour chaque sommet de la géométrie des objets, en fonction d'un ensemble de paramètres qui varient selon la méthode utilisée (accessibilité, exposition au soleil, température, etc.). La reconstruction de la surface de neige est réalisée en copiant la géométrie d'origine et en la déplaçant en fonction de cet ensemble de vecteurs.



FIGURE 1.26 – Génération de neige par déplacement de surface en deux étapes.

Foldes a proposé une méthode créant de grandes couches de neige, utilisée pour enneiger des collines ou des montagnes [47]. Cette méthode fonctionne en deux étapes et se base sur l'accessibilité locale du terrain (comme la présence de trous et de fossés) et l'énergie solaire. La première étape consiste à calculer, en chaque point du maillage représentant le terrain, l'accessibilité (stockée sous forme de texture de luminance) qui détermine la quantité de neige qui peut s'accumuler sur ce point. Le rôle de la seconde étape est de dissiper la neige accumulée précédemment lors de la phase de sublimation causée par la chaleur solaire. La précision de cette méthode est directement liée à la précision du maillage de terrain, et elle est utilisée pour créer une couche de neige faite pour être visualisée à une grande distance. Cependant, cette méthode est celle qui permet de produire les plus grandes scènes enneigées de cet état de l'art, et elle a l'avantage d'être utilisable très facilement sur les cartes de hauteurs qui sont un modèle numérique de terrain extrêmement répandu (Figure 1.27).

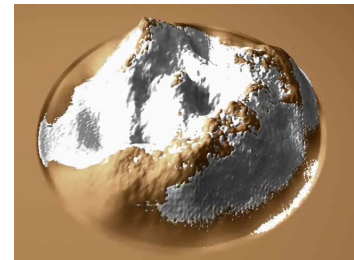


FIGURE 1.27 – Exploitation de l'accessibilité [47]

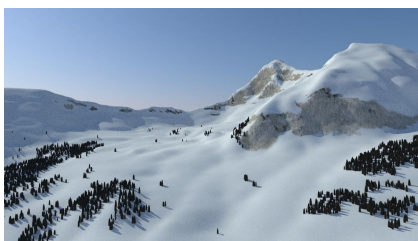


FIGURE 1.28 – Analyse d'image [112]

La méthode proposée par Premoze [112] est, comme la précédente, une méthode faite pour afficher de grands terrains vus de loin ; mais elle est basée sur l'analyse d'images panchromatiques aériennes (Figure 1.28). Une première phase de la méthode classe les différentes parties de l'image selon la nature du terrain (neiges, neiges ombragées, arbres, etc.). En utilisant les informations liées à la direction de la lumière, un prétraitement est fait sur les zones ombragées pour supprimer les ombres existantes au moment de la prise de photo, afin de pouvoir calculer l'éclairage de la scène à n'importe quel moment de la journée. La phase suivante consiste à simuler la création de la couche de neige en utilisant une carte de hauteurs divisée en une grille dont chaque cellule stocke la quantité de neige. La quantité de neige calculée en chaque point du maillage tient compte des précipitations, de la fonte de la neige et de l'ablation, qui correspond à la quantité de matière perdue par décomposition chimique lors des changements d'état.

Contrairement aux techniques présentées précédemment, la méthode de Ohlsson [100] utilise les capacités des processeurs graphiques, à travers une exploitation de mécanismes similaires à ceux utilisés dans le calcul de la projection d'ombres (Figure 1.29). Cette approche se divise en deux étapes : sélectionner les zones susceptibles d'être enneigées, puis déterminer la quantité de neige qu'elles peuvent recevoir. La sélection des zones sur lesquelles la neige va se déposer se fait en fonction de l'inclinaison des faces et de l'exposition au soleil. Ensuite, la neige accumulée est représentée avec un shader qui déplace les sommets du maillage d'origine. Cependant, la quantité de neige qui s'accumule sur chaque face ne dépend pas de l'aire de la face correspondante, ce qui crée des incohérences visuelles, comme une accumulation de neige trop importante sur des faces horizontales à la surface très faible. De plus, cette méthode ne fonctionne pas correctement dans le cas de modèles géométriques complexes qui contiennent des zones concaves, puisque le maillage de neige créé ne sera pas manifold et interpénétrera le modèle d'origine.

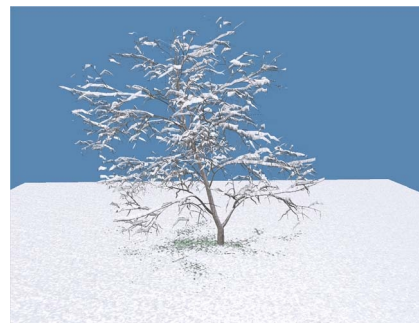


FIGURE 1.29 – Génération de neige sur processeur graphique [100]

Génération de neige à partir de piles de matière

Les méthodes de génération de neige basées sur des piles de matières utilisent une subdivision de l'espace selon une grille régulière en deux dimensions comme illustré dans la figure 1.30. De la neige s'accumule dans les différentes piles de la grille au cours de la simulation. Suivant les méthodes, différents tests ont lieu à chaque itération de la simulation, afin de prendre en compte différents phénomènes : chute de la neige d'une pile haute vers une pile plus basse, changement de phase de la neige lié à la température, etc. La surface de neige finale est ensuite extraite grâce aux informations de hauteur contenues dans les différentes piles.



FIGURE 1.30 – Génération de neige à partir de piles de matière.

Marechal est le seul à avoir présenté une réelle simulation volumique de la neige [77]. Cette méthode simule la chute de neige sur le sol et son évolution en tenant compte des données suivantes : quantité de neige, conduction, convection et radiation thermique, couverture nuageuse, point de rosée et cycle jour-nuit. Les changements de phase, tels que la fonte de la neige qui se transforme en eau, sont également gérés. La méthode repose sur un modèle voxelique : il est donc d'abord nécessaire de voxelliser le terrain. Ensuite, la boucle de simulation se déroule de la manière suivante : accumulation de neige en fonction des paramètres environnementaux, calcul des transferts thermiques entre les voxels, changements de phase des voxels et finalement mise à jour des paramètres environnementaux. Cette méthode produit de très bons résultats : elle permet de constater la fonte de la neige devenant de l'eau, ou la glaciation d'étendues d'eau en fonction de leur exposition au soleil (Figure 1.31). Cependant, elle repose sur une discrétisation

de l'espace, ce qui la rend extrêmement coûteuse en terme de calculs, et implique que la précision des scènes produites est inversement proportionnelle à leur taille.



FIGURE 1.31 – Trois étapes de la glaciation d'un lac [77].

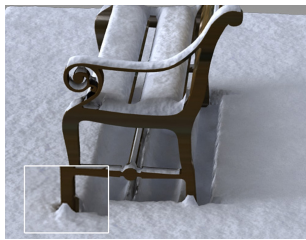


FIGURE 1.32 – Piles de neige [134]

Festenberg et Gumhold ont proposé un modèle statistique phénoménologique de modélisation de neige basé sur des observations réelles [134]. Cette approche est centrée sur la distribution de neige dans une scène, et ne prend pas en compte la chute des flocons de neige. Le cœur de la méthode repose sur une division de l'espace en une grille régulière, afin d'obtenir une carte de hauteurs. Chaque pile de la grille connaît les piles avec lesquelles elle peut échanger de la matière, et la neige est distribuée sur les piles en fonction de l'accessibilité. L'étape suivante consiste à stabiliser les piles obtenues, en exploitant les transitions entre les piles. Les piles sont ensuite regroupées par hauteur, pour créer des parcelles de neige. Finalement, un calcul sur les limites extérieures des parcelles permet de déterminer le profil de la couche de neige sur les bords. Cette méthode permet dans certaines conditions de produire des résultats présentant des caractéristiques intéressantes, comme l'accumulation de neige contre les surfaces verticales (Figure 1.32). Cependant, elle présente plusieurs inconvénients : il est nécessaire que les objets soient alignés avec la grille pour produire des résultats exploitables, et même dans ce cas, des problèmes de précision apparaissent, liés à la taille des objets et de la grille. Elle partage également avec les autres méthodes basées sur une division de l'espace un inconvénient majeur : elle ne peut produire que de petites scènes, et la précision des scènes est inversement proportionnelle à leur taille.

Une amélioration de cette méthode été proposée par les même auteurs deux ans plus tard [135]. Les transitions entre piles de neige ainsi que le calcul du profil de la neige sont remplacés par l'utilisation de noyaux de diffusion de neige. Pour un site de dépôt de neige donné, le calcul effectué avec les noyaux de diffusion prend en compte les sites voisins. Cette dépendance entre les sites permet de créer des ponts de neige, une caractéristique de la neige qui n'est à notre connaissance reproduite par aucune autre méthode. De plus, l'affichage de la couche de neige modélisée est plus élaboré que dans la plupart des autres méthodes et utilise une texture partiellement transparente faite pour boucler, afin d'afficher les bords des surfaces générées avec plus de réalisme (Figure 1.33). Cette amélioration est capable de produire des résultats très convaincants, même si elle souffre toujours des inconvénients liés à la division de l'espace.



FIGURE 1.33 – Piles de neige améliorées [135]

Autres méthodes liées à la neige

D'autres travaux scientifiques centrés sur la neige, sans traiter directement la création du manteau neigeux, ont été publiés. Le modèle de matériau proposé dans [127] permet d'animer des matériaux granuleux tels que le sable ou la neige en utilisant une carte de hauteurs. Dans [147], le manteau neigeux au sol est créé à partir de cartes de déplacements, tandis que les arbres enneigés sont rendus à l'aide d'une texture volumétrique. Langer a présenté une méthode hybride de rendu de chute de neige en combinant un système de particules avec une texture créée à partir d'analyse spectrale [73]. Récemment, Stomakhin a proposé une méthode d'animation de neige basée sur un modèle élastoplastique continu [125]. Cette simulation est finement configurable et permet de simuler la neige dans différents états. Schäfer a présenté une méthode de déformation de surface de subdivision en temps réel sur processeur graphique pour représenter différents types de traces dans la neige [120].

Synthèse

Les méthodes de génération de neige souffrent toutes de la même limitation : elles ne fonctionnent que sur des plages d'échelles limitées, comme le montre la figure 1.34. La méthode [47] est conçue pour visualiser de grands terrains couverts de neige observés à grande distance. À l'autre extrémité du spectre, [60] est capable de produire des détails très fins, inférieurs au centimètre, mais elle ne fonctionne que sur des scènes inférieures au mètre. On constate que le facteur d'échelle, défini comme le rapport entre la taille des scènes et la taille des détails, ne dépasse jamais 10^3 , quelle que soit la méthode. Il est par exemple impossible de générer une couche de neige à l'échelle d'une ville en conservant des détails de l'ordre du centimètre. Il y a un réel besoin pour des méthodes avec un facteur d'échelle supérieur à 10^3 , permettant à la fois de générer une très grande couverture neigeuse tout en conservant un niveau de détails très fin.

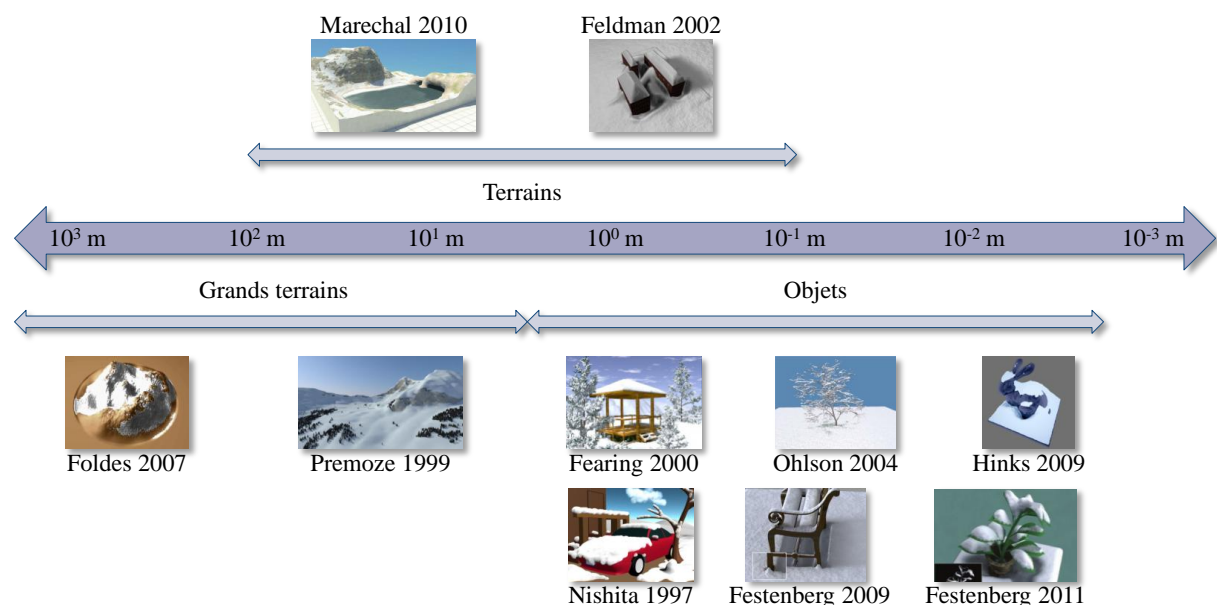


FIGURE 1.34 – Différentes méthodes de modélisation de neige classées en fonction de l'échelle des scènes qu'elles sont capables de produire : grands terrains, terrains et objets.

Les différentes méthodes souffrent également de limitations particulières, souvent liées aux structures de données discrètes servant à représenter la neige. Les méthodes utilisant une subdivi-

vision régulière de l'espace sont très dépendantes de la précision utilisée. Plus la subdivision est précise, plus leur empreinte mémoire et leurs temps de calculs augmentent. Cet impact est tel que la précision est un facteur bloquant, rendant impossible l'utilisation de méthodes utilisant des structures discrètes avec des résolutions très fines. De plus, même si ce type de méthode est bien adapté à des terrains, il pose différents problèmes lorsqu'il est utilisé avec des objets manufacturés, tels que du mobilier par exemple. La figure 1.35 met en évidence les différents cas rencontrés avec ces méthodes. Les trois premiers cas montrent un objet simple modélisé à base de plans et d'angles droits, que l'on rencontre régulièrement dans les mondes virtuels. Quand les dimensions de l'objet sont un multiple des dimensions de la grille, et que l'objet est aligné sur la grille, comme dans le cas n°1, la configuration est optimale, et la méthode fonctionnera au mieux. Cependant, ce cas idéal n'a concrètement aucune chance de se produire. Dans le cas n°2, l'objet est aligné sur la grille, mais ses dimensions ne sont pas un multiple de celles de la grille. Ce cas peut se rencontrer fréquemment dans une scène. La méthode produira alors des artefacts aux bords de l'objet dans les cellules rouges, mais ils peuvent être acceptables selon les configurations. Le cas n°3 est pire : l'objet n'est pas aligné sur la grille. Les artefacts produits dans les cellules rouges créeront cette fois-ci un crénelage, rendant le résultat très mauvais. Dans le cas n°4, l'objet que l'on veut recouvrir de neige est plus complexe. Là aussi, de nombreux artefacts impliquent un crénelage important, créant un résultat de piètre qualité. Le seul moyen d'atténuer l'influence de ces artefacts est d'opter pour une subdivision de l'espace extrêmement fine, ce qui augmente de façon drastique la consommation de mémoire et les temps de calcul et limite la taille des scènes que la méthode peut produire.

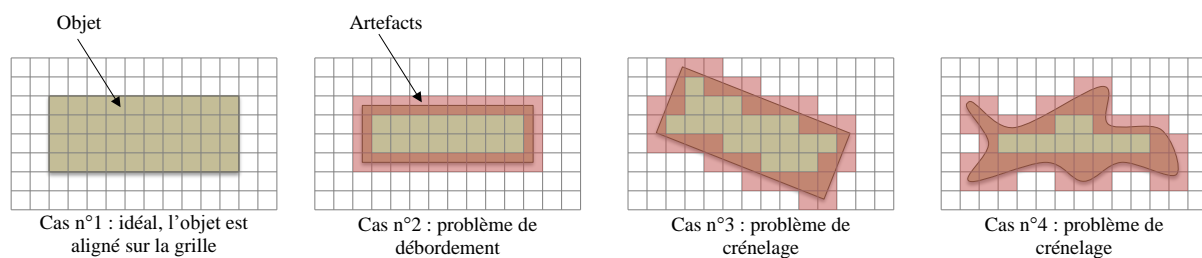


FIGURE 1.35 – Les différents cas rencontrés avec les méthodes de génération de neige reposant sur la subdivision de l'espace en grille régulière.

4.1.2 Génération de glace

Les techniques de génération de glace se décomposent en deux catégories : les méthodes s'inspirant de techniques de simulation physique pour la génération de givre, et les techniques de construction de stalactites de glace.

Génération de givre

La génération de givre repose sur des simulations de croissance de cristaux (Figure 1.36). À partir d'un ou de plusieurs sites de départ et en fonction de différents facteurs tels que la température et l'humidité, les cristaux de givre se développent, se diffusent et colonisent itération après itération la surface de l'objet.

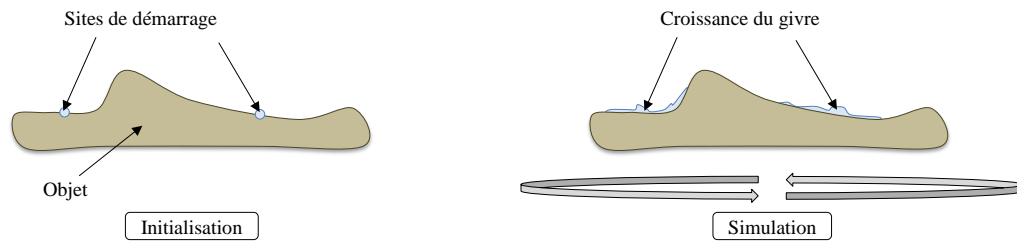


FIGURE 1.36 – Développement des cristaux de givre à partir de sites de démarrage.

Kim a proposé une méthode de simulation de croissance de cristaux de glace [71]. Ces travaux sont basés sur un modèle à champ de phase, un outil utilisé en physique numérique pour résoudre les problèmes d'interface entre différents domaines. L'implémentation est réalisée sur processeur graphique afin d'améliorer les temps de calcul, et les auteurs introduisent un nouvel algorithme géométrique qui permet d'améliorer les résultats du modèle implicite de la méthode, qui produit à l'origine des résultats trop lisses. Les temps de calcul de cette simulation sont plutôt courts, puisque les différents exemples sont calculés en environ une minute. Cependant, elle ne fonctionne que dans un contexte précis : elle ne peut être utilisée que pour recouvrir de glace des objets dont les surfaces sont faibles, puisque de l'ordre du mètre carré au maximum (Figure 1.37).

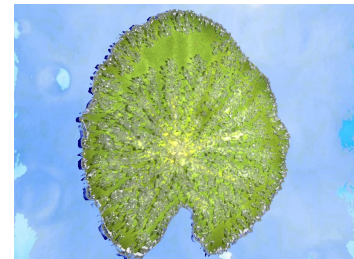


FIGURE 1.37 – Croissance de cristaux de glace [71]

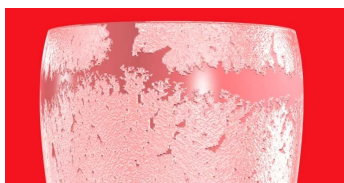


FIGURE 1.38 – Formation de givre [70]

L'année suivante, Kim a présenté une amélioration de sa méthode [70]. Un solveur de fluide et une méthode d'agrégation limitée par diffusion sont utilisés pour améliorer la simulation originale. Le modèle proposé repose sur une approche physique inspirée par les réactions thermodynamiques intervenant lors de la formation de la glace. Les temps de calcul de cette méthode sont du même ordre de grandeur que la précédente, mais les résultats qu'elle produit sont considérablement meilleurs (Figure 1.38).

Construction de stalactites de glace

Les méthodes de génération de stalactites de glace reposent sur l'utilisation de particules. Elles sont divisées en deux étapes (Figure 1.39). La première étape est une boucle de simulation qui consiste à déplacer des particules représentant les gouttes d'eau qui ruissellent, que ce soit à la surface de l'objet, ou à la surface des particules qui se sont précédemment agrégées sur l'objet. Le déplacement des particules dépend de facteurs variant selon les méthodes : température, masse, etc. La seconde étape consiste à construire la surface de la stalactite à partir de l'ensemble des particules agrégées.

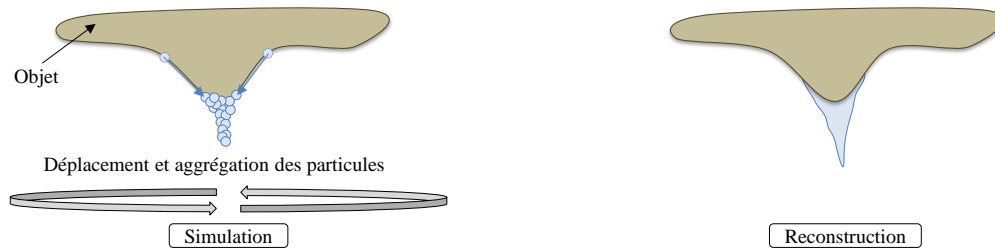


FIGURE 1.39 – Les deux étapes de la génération de stalactites de glaces.

Kharitonsky et Gonczarowski ont proposé la première des méthodes de génération de stalactite de glace [68]. Leur modèle physique est basé sur les processus mécaniques et thermodynamiques qui interviennent lors de la formation des stalactites. Le déplacement des gouttelettes d'eau est calculé le long de la surface des stalactites en tenant compte de leur masse, de leur vitesse, de leur température, de la gravité et des forces de tension superficielle. La solidification des gouttelettes d'eau est calculée à partir du transfert thermique entre les différents matériaux et de leur conductivité thermique. La surface des stalactites est ensuite convertie en carreaux de Bézier pour être affichée par lancer de rayons (Figure 1.40).



FIGURE 1.40 – Génération de stalactites de glace [68]

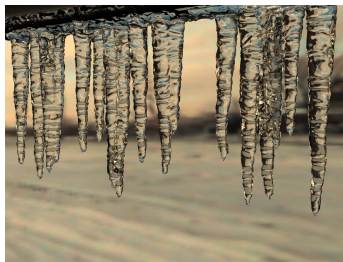


FIGURE 1.41 – Solidification de fines couches d'eau [69]

Kim a également proposé une méthode de génération de stalactites [69]. Elle repose en premier lieu sur une variante nommée *couche-fine* du problème de Stephan, utilisée conjointement à des lignes de niveau pour résoudre les problèmes de solidification à très petite échelle. Ce modèle gère la modélisation grossière de la stalactite, mais il produit un résultat trop lisse. Afin d'améliorer le résultat, une technique inspirée par la physique est ensuite utilisée pour calculer les ondulations qui se forment le long des stalactites (Figure 1.41).

Contrairement aux modèles précédents, celui qui est exposé dans [48] permet, en plus de la création de stalactites, la modélisation d'une couche de glace qui englobe partiellement ou entièrement un objet. La méthode est divisée en quatre étapes : pour chaque sommet du maillage de l'objet, calcul du coefficient d'eau (quantification de l'accessibilité de ce point par l'eau), détermination des points d'écoulement (points de départ des stalactites), calcul de la trajectoire des stalactites et finalement calcul de la surface de la glace. Un utilisateur peut intervenir après chaque étape pour modifier les résultats produits par la dernière étape. Ce modèle gère les collisions et permet de créer des stalactites sophistiquées et cohérentes, en contact avec la surface de plusieurs objets, et il fournit des résultats de très bonne qualité (Figure 1.42).



FIGURE 1.42 – Génération d'une couche de glace [48]

La méthode utilisée dans [128] utilise un modèle de marche aléatoire afin de calculer le déplacement des gouttelettes couplé à une grille régulière tridimensionnelle pour générer des

stalactites. Dans [119], la trajectoire des gouttelettes d’eau qui forment une stalactite est calculée en utilisant la méthode *SPH*, un outil très utilisé en simulation de fluides.

4.1.3 Synthèse

La modélisation de phénomènes hivernaux a été étudiée au travers d’approches très diverses. Il existe des méthodes pour créer de la neige ou de la glace à des échelles très différentes : certaines peuvent couvrir plusieurs kilomètres de terrain alors que d’autres sont faites pour modéliser finement ces phénomènes au niveau d’un seul objet, à une échelle inférieure au centimètre. Les méthodes qui produisent les meilleurs résultats, que ce soit pour la neige ([60] et [77]) ou pour la glace ([48]), sont des simulations qui prennent en compte différents phénomènes physiques complexes de manière réaliste. Cependant, ce sont les méthodes les plus difficiles à mettre en œuvre, et les plus coûteuses en temps de calcul.

4.2 Phénomènes végétaux

4.2.1 Feuilles

Pour distribuer des feuilles dans une scène, trois étapes sont nécessaires (Figure 1.43). Il faut d’abord modéliser les feuilles puis ensuite les faire vieillir avant de les distribuer. En pratique, les deux premières étapes sont souvent gérées au sein des mêmes méthodes, puisque la modélisation et le vieillissement nécessitent en général des structures de données identiques et sont fortement corrélés, au point que le vieillissement est généralement pensé comme une paramétrisation de la modélisation des feuilles. La distribution peut par contre être totalement dissociée des deux étapes précédentes, et il est possible de générer des feuilles à partir d’une méthode puis de les distribuer avec une autre.

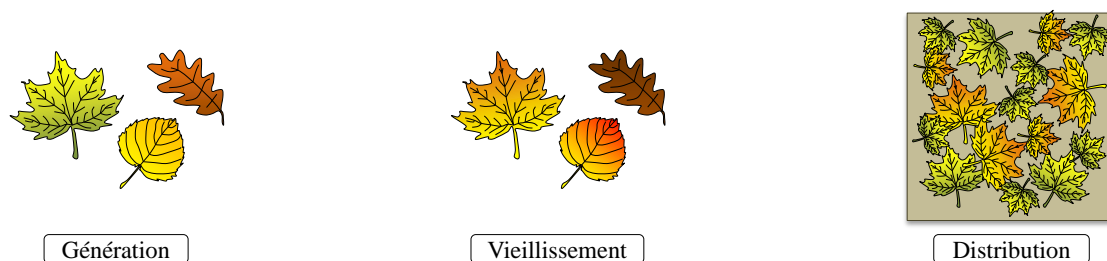


FIGURE 1.43 – Les trois étapes de la génération de feuilles au sol.

Modélisation et vieillissement de feuilles et de feuillages

Des méthodes fondamentalement différentes ont été proposées pour générer de manière réaliste différentes espèces de feuilles. Mündermann a mis au point une méthode de création de feuilles reposant sur un squelette créé à partir d’un diagramme de Voronoï [93]. Peyrat utilise un *L-System* paramétré et des cartes généralisées à deux dimensions pour créer des ensembles de feuilles d’une même essence, en faisant varier leur texture et leur géométrie [107]. En prenant en compte le vieillissement et la dégradation naturelle des feuilles, cette méthode produit des résultats qui augmentent significativement le réalisme d’une branche en la peuplant de feuilles semblables mais uniques. Ce modèle gère également des caractéristiques de haut niveau sous la forme de trous créés par les insectes qui se



FIGURE 1.44 – Modélisation de feuilles par *L-system* et carte généralisée [107]

nourrissent des feuilles (Figure 1.44).



FIGURE 1.45 – Structure duale [66]

Jeong utilise une structure duale, composée d'une triangulation de Delaunay et du diagramme de Voronoï associé pour modéliser des feuilles [66]. Un modèle biologique associé à cette structure de données simule l'évaporation de l'eau, qui est la cause principale du racornissement des feuilles. La modélisation complète d'une feuille requiert un temps de calcul de l'ordre de la seconde, et cette méthode produit des résultats de très bonne qualité (Figure 1.45).

D'autres méthodes se sont focalisées sur la génération réaliste du feuillage des arbres, dont l'apparence varie au fil du temps. Dans [31], le phototropisme et l'évolution des pigments des feuilles sont utilisés pour définir l'apparence du feuillage. Braitmaier a proposé de déterminer la couleur des feuilles en se basant sur l'évolution au court des saisons de la concentration des quatre pigments les plus importants [21]. Une autre méthode semblable, basée sur l'influence du climat, détermine l'évolution des feuilles en utilisant les informations relatives aux pigments responsables des couleurs verte, jaune et rouge des feuilles [149].

Distribution de feuilles

Desbenoit *et alii* sont les seuls à s'être penchés sur la distribution de grandes quantités de feuilles mortes dans le but de créer des scènes automnales [36]. Cette méthode est divisée en deux parties distinctes : les feuilles sont d'abord générées en fonction de l'environnement, et elle sont ensuite distribuées dans la scène. Les feuilles sont modélisées à partir de feuilles réelles qui ont été scannées, afin de capturer toute la complexité de leur texture. Un système de chaînes de Markov pilote l'évolution des feuilles en fonction de l'humidité et de la température. La seconde étape de la méthode consiste à simuler la position que les feuilles qui se détachent des branches vont atteindre à l'aide de trajectoires procédurales types et de détection de collisions (Figure 1.46). Cependant, cette méthode ne fonctionne pas à grande échelle : elle ne peut distribuer plus de 25 000 feuilles, et la taille des scènes est limitée à quelques centaines de mètres carrés.



FIGURE 1.46 – Trajectoires procédurales [36]

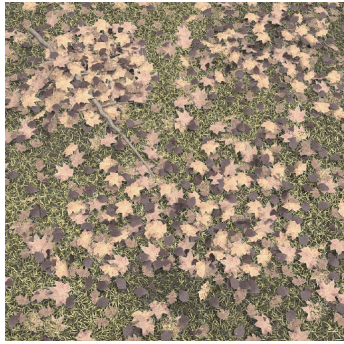


FIGURE 1.47 – Distribution de feuilles dans un modèle d'écosystème [38]

Le modèle d'écosystème complexe présenté dans [38] peut distribuer des feuilles mortes en plus de la végétation. Cependant, cette méthode ne permet d'en distribuer que de petites quantités, et elle ne peut pas générer de distribution complexe : le tas de feuilles autour de la branche visible dans la figure 1.47 a été créé manuellement. Des techniques ont également été mises au point pour animer des feuilles transportées par le vent. Wejchert et Haumann ont utilisé la mécanique des fluides et des primitives de flux pour créer des flux d'air capables de déplacer et de transporter des feuilles [141]. Wei a mis au point une modélisation de flux complexes, capable de transporter des feuilles et d'autres objets tout en les déformant de manière réaliste [140].

Synthèse

Les méthodes de génération de feuilles sont nombreuses et variées. Elles permettent de générer des feuilles de différentes essences et à différentes étapes de leur vie, que cela soit à l'échelle des feuilles en elles-mêmes, ou à l'échelle du feuillage entier d'un arbre. Cependant, il y a un réel manque au niveau de la distribution de feuilles dans une scène. Seul [36] traite cette problématique, et le problème reste ouvert, puisque la taille des scènes et le nombre de feuilles qu'il est possible de générer sont limités : environ 25 000 feuilles au total, pour une scène qui ne dépasse pas quelques centaines de mètres carrés. Il est par exemple impossible de distribuer automatiquement des feuilles mortes à l'échelle d'une ville ou même d'un quartier, alors que ce phénomène fait drastiquement évoluer l'apparence des villes au fil des saisons. Il est également impossible de distribuer des feuilles automatiquement sur de grandes scènes naturelles, alors que c'est un phénomène majeur dans le changement d'apparence de ces zones.

4.2.2 Herbe

La modélisation d'herbe a été étudiée selon trois angles complémentaires : distribution, animation et rendu. La distribution d'herbe a principalement été abordée en commun avec la distribution de végétaux de tous types, dans les méthodes de simulation d'écosystèmes. Elle nécessite deux étapes : il est d'abord nécessaire de modéliser les brins d'herbe avant de pouvoir les distribuer (Figure 1.48). Les méthodes d'animation d'herbe se sont intéressées aux conséquences du vent et des différents flux d'air sur des surfaces de quelques centaines de mètres carrés au maximum. Finalement, le rendu de l'herbe a été analysé sur des surfaces plus grandes, et majoritairement à travers un ensemble de techniques fonctionnant en temps réel.



FIGURE 1.48 – Les deux étapes de la génération d'herbe au sol.

Distribution d'herbe

Les simulations d'écosystèmes tendent à reproduire de manière réaliste la distribution de plusieurs espèces de végétaux en se basant sur la compétition pour l'accès à différents types de ressources qui leur sont nécessaires, tel que la lumière, l'eau, l'espace ou les différents éléments qu'elles tirent du sol pour se nourrir.



FIGURE 1.49 – Écosystème complexe [38]

Le premier modèle d'écosystème a été proposé par Deussen *et alii* dans [38] et permet de créer des scènes complexes comprenant des milliers de plantes. La distribution originale des plantes est définie par l'utilisateur à l'aide de différentes cartes en niveaux de gris qui définissent leurs propriétés (espèce, âge, etc.). Lors de la simulation, un modèle biologique complexe guide l'évolution des plantes, en déterminant quels sont les individus de la population qui naissent, se développent et meurent. Lors de l'affichage de l'écosystème, la population de plantes est trop importante pour générer chaque individu séparément. Des hiérarchies d'instances sont créées, et les individus qui partagent des paramètres proches sont représentés par la même instance. Cette méthode permet d'obtenir de bons résultats visuels (Figure 1.49). Cependant, le réalisme biologique de ces résultats est discutable à cause du faible nombre de paramètres gérés par le modèle d'évolution biologique.

Le modèle d'écosystème proposé par Benes se focalise sur l'équilibre entre la distribution spatiale des plantes et leur densité [9]. Il fait partie des modèles d'écosystèmes désignés comme *local vers global*, qui simulent la compétition pour les ressources des végétaux. À chaque itération de la simulation, les plantes peuvent disséminer des graines, les plantes obsolètes (trop vieilles, qui entrent en collision, etc.) sont éliminées, et les plantes restantes se développent. Ce modèle permet d'itérer sur des durées importantes, de l'ordre de plusieurs années, et d'atteindre un état stable. Cependant, bien que l'intégration de nouvelles espèces de végétaux soit décrite comme facile, il n'en gère que deux, et les résultats produits occupent une surface de seulement quelques mètres carrés (Figure 1.50).



FIGURE 1.50 – Écosystème évoluant sur le long terme [9]



FIGURE 1.51 – Compétition asymétrique de végétaux [3]

La méthode de simulation d'écosystème présentée dans [3] est plus complexe que les deux précédentes et vise à fournir des résultats plus proches de la réalité en introduisant une distinction des modes de compétition pour les ressources. Quand deux plantes de taille identique sont en compétition dans un même espace, elles sont en *compétition symétrique*, et leurs comportements sont identiques. Comme elles reçoivent les mêmes quantités de ressources, elles grandissent de façon homogène. Au contraire, lorsque les plantes en compétition ne sont pas de même taille, la compétition est désignée comme *compétition asymétrique*, et les plus imposantes absorberont plus de ressources, ce qui leur permettra de se développer plus vite. En conséquence, le manque de disponibilités de ressources pour les plantes les plus petites les contraindra à se développer moins vite. Les résultats mettent clairement en évidence ce phénomène, et l'on voit apparaître dans l'écosystème final les zones où les végétaux ne peuvent pas

se développer, à cause du manque de ressources lié à la compétition importante créée par les végétaux environnants (Figure 1.51).

D'autres techniques variées ont été proposées pour distribuer des végétaux. Dans [10], le développement de l'écosystème est guidé par un ensemble d'agents autonomes. Les groupements de plantes d'une même espèce ont été calculés à l'aide de courbes implicites dans [11]. Les *L-system multiset* peuvent également être utilisés pour modéliser les groupements de plantes [72].

Rendu d'herbe

Le principal verrou technologique rencontré lors du rendu d'herbe est la complexité des scènes, puisque la densité de l'herbe est très élevée : environ 100.000 brins par mètre carré. Il est donc très difficile d'effectuer le rendu en temps réel d'une grande scène d'herbe où chaque brin est modélisé de façon classique à l'aide de plusieurs polygones, puisque cela revient à traiter en temps réel une quantité trop élevée de polygones.

La méthode proposée dans [37] vise à améliorer le rendu de grandes scènes complexes de végétation. Le rendu classique peut rapidement être laborieux pour des scènes de végétation complexes, puisqu'elles sont composées de milliers de végétaux eux-mêmes composés de milliers de polygones. Afin de ne pas être limité par le nombre de polygones de telles scènes, les modèles de végétaux sont convertis en un format qui combine des polygones, des points et des lignes ; et ce pour plusieurs niveaux de détails. Cette phase de pré-calcul est une phase de décimation d'informations, et elle peut prendre en compte l'importance, afin de réduire l'information de façon hétérogène dans le but de ne pas supprimer les caractéristiques intéressantes des végétaux. Le rendu final est effectué grâce à un mélange des différentes représentations des objets (Figure 1.52).



FIGURE 1.52 – Temps réel, 120 millions de triangles [37]



FIGURE 1.53 – Temps réel avec éclairage dynamique [20]

Boulangier [19] s'est intéressé au rendu de scènes naturelles en temps réel avec un éclairage dynamique. Une partie de ses travaux en particulier est centrée sur le rendu d'herbe [20]. Dans cette méthode, la distribution de l'herbe est pilotée par une carte de densité. Le terrain contenant l'herbe est découpé en trois types de cellules qui utilisent chacune un type de contenu, en fonction de la distance de la caméra. Les cellules les plus proches de la caméra contiennent de la géométrie, les cellules intermédiaires contiennent un ensemble de plans horizontaux et verticaux semi-transparents entremêlés, et les cellules les plus éloignées de la caméra ne contiennent qu'un seul plan horizontal texturé. Dans les zones de transitions entre différents types de cellules, les deux modes de rendu concernés sont pondérés et utilisés simultanément, assurant une transition sans artefact visuel. Au final, le nombre d'images par seconde est très dépendant du point de vue, mais l'affichage reste suffisamment fluide pour du temps réel et les images produites sont de très bonne qualité (Figure 1.53).

Récemment, Jahrman et Wimmer ont présenté une technique de rendu d'herbe exploitant les capacités de tessellation des processeurs graphiques [65]. Contrairement à la très grande majorité des autres méthodes qui mélangent différents types de représentation, l'intégralité de l'herbe est dans ce cas affichée sous forme de géométrie. Deux cartes de densité sont utilisées pour définir la façon dont l'herbe recouvre le terrain : la première définit la densité et la hauteur de l'herbe, et la seconde définit son apparence et sa composition. Les brins d'herbe sont d'abord créés sous forme de rectangle, puis successivement déformés, subdivisés et colorés. Une carte de force permet de représenter l'herbe ayant subi des contraintes externes, par exemple les traces laissées par un humain qui s'y serait déplacé. Une fonction de vent est utilisée pour animer les brins d'herbe. Cette technique est optimisée avec de l'instanciation, afin de ne pas avoir un nombre trop élevé de brins d'herbes différents à gérer, et avec une fonction de niveau de détail, afin de ne pas générer une géométrie complexe pour les brins situés loin de la caméra. Les résultats produits sont de bonne qualité, et le nombre d'images par seconde est élevé, même dans de grandes scènes contenant jusqu'à un million de brins d'herbe (Figure 1.54).

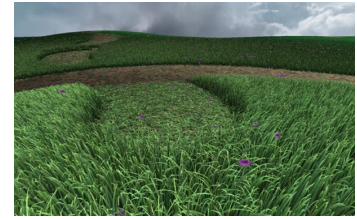


FIGURE 1.54 – Temps réel, herbe entièrement polygonale [65]

D'autres méthodes hétérogènes ont été proposées pour calculer le rendu de l'herbe en temps réel. La technique utilisée dans [23] repose sur une structure discrète de terrain. Seules les cellules de la structure qui sont dans le cône de vision sont générées puis affichées, afin de limiter la quantité de données à gérer. La méthode présentée dans [105] décrit comment rendre efficacement de grandes surfaces d'herbe, en utilisant de grands polygones et une texture semi transparente. Des images pré-calculées de modèles d'herbe sont exploitées pour créer une fonction de texture bidirectionnelle utilisée pour afficher l'herbe en temps réel [121]. Les *billboards volumétriques* permettent d'afficher efficacement différents types d'objets, y compris de la végétation en général - et de l'herbe en particulier [33]. Zhao [148] propose de n'utiliser que deux niveaux de détails et d'animer l'herbe avec des calculs de collision entre la géométrie des brins d'herbes et d'autres objets. Le processeur graphique est également utilisé pour calculer le raffinement adaptatif des *billboards* d'herbe déformés par des forces extérieures [101].

Animation d'herbe

La première problématique liée à l'animation de l'herbe est très proche de celle liée au rendu d'herbe : la quantité de données est un obstacle, et comme il faut afficher l'herbe une fois son animation calculée, il faut que les structures de données utilisées soient adaptées aux deux types de calculs. La seconde problématique est le contrôle de l'animation : les mouvements de l'herbe sont principalement causés par le vent et les flux d'air, et il est nécessaire d'établir un modèle qui puisse les décrire efficacement.

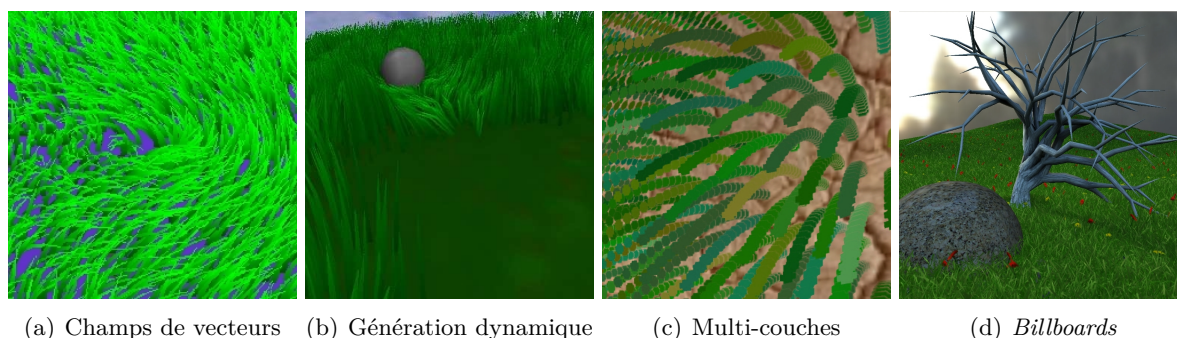


FIGURE 1.55 – Animation d'herbe. De gauche à droite : primitives de vent modélisées par des champs de vecteur [106], primitives de vent améliorées et génération dynamique de l'herbe [57], structures multi couches [5], et modélisation par *billboards* [58].

Perbet [106] utilise des primitives de vent pour définir les mouvements de l'herbe. Ce sont des champs de vecteurs qui se déplacent à la surface du terrain. Les primitives de vent sont variées (brises, rafales, tornades, etc.) et plusieurs primitives peuvent se déplacer simultanément dans une prairie et se croiser, mais leurs effets ne se cumulent pas (Figure 1.55(a)). Cette approche a été étendue en modélisant dynamiquement l'herbe lors de l'affichage à partir d'une carte de densité [57]. La base du système d'animation est commune à la méthode dont elle est inspirée : des primitives d'animation se déplacent à la surface du terrain et envoient des informations qui pilotent le comportement de l'herbe, telles que l'amplitude du vent ou la direction dans laquelle les brins d'herbe doivent se courber (Figure 1.55(b)). Bakay [5] a présenté une méthode d'animation en temps réel combinée à une méthode de rendu spécifique. Des couches d'herbe, copies du terrain sur lequel se développe l'herbe, sont affichées depuis le bas vers le haut en transparence, à l'exception des zones intersectées par un brin d'herbe (Figure 1.55(c)). L'animation est réalisée en déplaçant les sommets des couches d'herbe en fonction d'un champ de vent. Une autre méthode représente l'herbe par un ensemble de *billboards* arrangés en grille régulière et texturés avec une texture semi-transparente [58]. Lors de rafales de vent, de larges surfaces d'herbe sont couchées uniformément au sol. Les turbulences dont les fréquences sont plus élevées causent des mouvements d'amplitude plus faible mais plus aléatoire. Au lieu de déplacer les *billboards*, une distorsion est effectuée lors de la recherche de la couleur des texels selon une carte de bruit (Figure 1.55(d)). Une approche physique [116] et un mélange d'approche physique et procédurale [117] ont également été utilisés pour animer l'herbe.

Synthèse

L'animation et le rendu d'herbe ont été largement couverts, et plusieurs méthodes fonctionnent sur de grandes surfaces en proposant des résultats convaincants. La distribution de l'herbe a été beaucoup plus étudiée que celle des feuilles, majoritairement par le biais des méthodes de simulation d'écosystèmes. Cependant, les simulations d'écosystèmes sont des méthodes complexes, et si elles permettent de reproduire assez fidèlement la réalité, c'est justement parce qu'elles y sont assujetties. Il n'existe pas de méthode permettant de contrôler facilement et intuitivement la distribution d'herbe.

De plus, la quasi-totalité des méthodes liées à l'herbe souffrent des mêmes inconvénients. L'herbe est modélisée par des brins espacés, alors qu'elle se développe en réalité sous forme de touffes qui regroupent des dizaines de brins. Un seul type de brin d'herbe est généralement utilisé pour modéliser des champs entiers, alors que de telles surface de terrain sont en réalité colonisées par plusieurs espèces de végétaux. Il n'existe pas non plus de méthodes permettant

de visualiser de l’herbe à différentes étapes de son développement, ou de l’herbe altérée par son environnement (grillée par le soleil, etc.).

5 Conclusion

Nous avons passé en revue dans ce chapitre un ensemble de méthodes utilisées pour générer du contenu graphique et principalement pour modifier l’aspect du contenu graphique existant, que ce soit au travers des méthode de vieillissement et de changements d’apparence, qui altèrent le contenu lui-même, ou des méthodes d’ajout de géométrie, qui rajoutent des détails géométriques à du contenu sans l’altérer.

Les techniques qui produisent les meilleurs résultats, quel que soit le domaine, sont les techniques de simulations. Parmi les simulations, celles qui prennent en compte le plus de phénomènes et de paramètres physiques sont celles dont le fonctionnement se rapproche le plus des phénomènes réels modélisés, et sont logiquement celles qui produisent les résultats les plus réalistes. Cependant, les techniques de simulation sont intrinsèquement sujettes à différents inconvénients. Elles sont coûteuses en calcul, lourdes en mémoire, et passent très mal à l’échelle : elles sont capables de capturer de très petits détails sur des scènes de taille réduite, mais elles sont incapables de conserver cette précision sur de grandes scènes. Elles sont également difficilement prédictibles : elles requièrent différents paramètres en entrée (température et quantité de précipitations pour une méthode de modélisation de neige par exemple), et il est difficile de définir quelle va être l’influence de chacun de ces paramètres sur le résultat produit. Il est ainsi nécessaire de répéter plusieurs fois une simulation avec des paramètres différents pour aboutir à un résultat proche de celui désiré, et il est nécessaire d’avoir des connaissances dans le domaine du phénomène modélisé pour comprendre comment et à quel point les paramètres vont influencer le résultat.

La plupart des méthodes d’ajout de géométrie passées en revue dans cet état de l’art souffrent du même inconvénient : elles sont difficilement contrôlables. Elles sont conçues pour gérer un phénomène à une échelle donnée, mais ne permettent pas, ou difficilement, d’effectuer des modifications locales. Il est donc difficile de les employer dans un contexte où l’importance du contrôle utilisateur sur le résultat produit est prépondérante.

À l’exception de [77] qui gère les différents changements de phase de l’eau et gère à la fois la neige et la glace, les méthodes d’ajout de géométrie ne traitent qu’un phénomène. Il est donc compliqué de représenter plusieurs phénomènes sur une même scène, d’autant plus que les formalismes des données nécessaires en entrée aux différentes méthodes sont parfois très éloignés. Il y a donc un réel besoin d’une méthode capable de générer simultanément différents phénomènes variés sur une même scène.

Dans le prochain chapitre, nous détaillons le fonctionnement et les mécanismes du nouveau modèle de génération de scène virtuelles développé lors de cette thèse, et nous détaillons le processus de construction des scènes.

Chapitre 2

Objets environnementaux



FIGURE 2.1 – Une grande scène créée à partir d’objets environnementaux. Elle varie du printemps, à gauche, jusqu’à l’hiver, à droite.

Table des matières

1	Introduction	41
2	Arbre de construction	41
2.1	Objets environnementaux	43
2.2	Opérateurs de combinaison	46
3	Évaluation des effets	47
3.1	Réaction à l’environnement	47
3.2	Interactions	49
4	Création des scènes	51
4.1	Base de données d’objets environnementaux	51
4.2	Création de l’arbre de construction	52
4.3	Génération de la scène décorée finale	54
5	Conclusion	54

1 Introduction

Dans ce chapitre nous présentons un modèle permettant de décorer des objets en leur ajoutant différents types de détails géométriques tels que de la neige ou des feuilles mortes. Notre approche repose sur un traitement spécifique par objet (Figure 2.2). Plutôt que d'utiliser une méthode de simulation globale pour tous les objets de la scène, chaque objet définit non seulement la manière dont il change d'apparence, mais également la manière dont il peut influencer l'apparence de ses voisins. Le contrôle des décorations se fait à l'aide de champs scalaires en trois dimensions, appelés champs d'influence. Ce sont des outils simples, efficaces, et hautement prédictibles, qui permettent aux utilisateurs de contrôler finement les différentes décorations avec une grande précision. Nous utilisons ce modèle pour décorer des scènes avec de la neige, de la glace, des feuilles mortes et de l'herbe. Les principales contributions de notre modèle sont les suivantes :

La variation d'échelle. Chaque type d'objet définissant la manière dont il change d'apparence, il est possible d'adapter la résolution des décorations géométriques produites lors du changement d'apparence à chaque objet tout en optimisant les traitements. Ceci permet la génération de grandes scènes de plusieurs milliers de mètres carrés comportant des détails très fins, de l'ordre du centimètre.

La vitesse. Notre modèle calcule l'apparence de chaque objet indépendamment, il ne nécessite donc pas de recalculer l'intégralité de la scène à chaque modification. De plus, notre approche par objet permet de paralléliser naturellement les calculs.

Le contrôle. Les champs d'influence sont intuitifs et prédictibles. Ils offrent la possibilité de contrôler précisément les décorations géométriques de la scène à l'échelle micro, qui correspond à des perturbations locales et à des détails de quelques centimètres (tel que de l'herbe grillée par la proximité d'une source de chaleur). De plus, très peu de paramètres sont nécessaires pour contrôler l'ensemble de la scène à l'échelle macro, c'est à dire pour des événements constants sur l'ensemble de la scène (comme par exemple les chutes de neige).

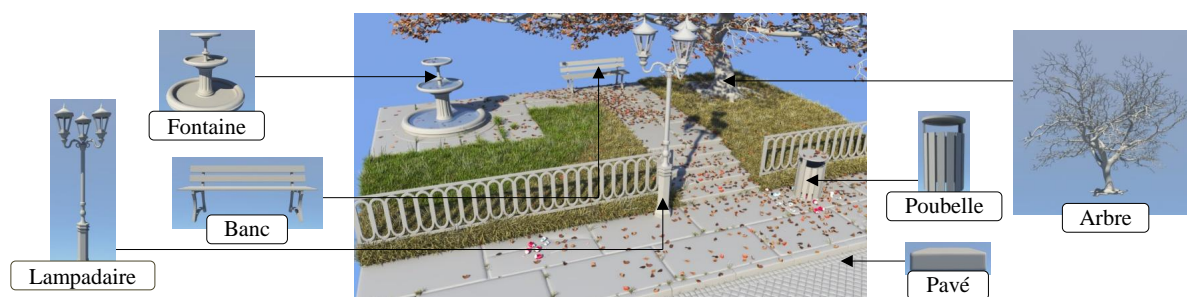


FIGURE 2.2 – Mise en évidence d'une partie des objets qui composent une scène décorée.

2 Arbre de construction

Nous proposons de représenter la scène sous la forme d'un arbre de construction hiérarchique dont les nœuds terminaux sont des *objets environnementaux* (section 2.1) et les nœuds internes des *opérateurs de combinaison* (section 2.2). Les objets environnementaux sont des objets géométriques augmentés par des *champs d'influence*, qui représentent leur impact sur l'*environnement*, et des *effets*, qui génèrent les décorations des objets sous la forme de détails géométriques. Ils

sont stockés dans une base de donnée, puis organisés sous la forme d'un arbre de construction, par génération procédurale ou par édition interactive. Lors de l'évaluation de l'arbre, les objets environnementaux se décorent et produisent des détails en fonction de l'environnement (Figure 2.3).

L'environnement, noté e , est composé de champs d'influence qui sont des champs scalaires en trois dimensions. Il contient l'ensemble des caractéristiques et propriétés météorologiques de la scène (chapitre 3). Il existe deux types de champs d'influence : les champs d'*effets* et les champs de *phénomènes*. Les effets réagissent à l'environnement et créent les décorations géométriques des objets. Ce sont la neige, la glace, les feuilles, et l'herbe. Leur fonctionnement est analysé en détail dans le chapitre 4. Les phénomènes modifient et perturbent la création de ces décorations. Ce sont l'accessibilité, la chaleur et l'humidité.

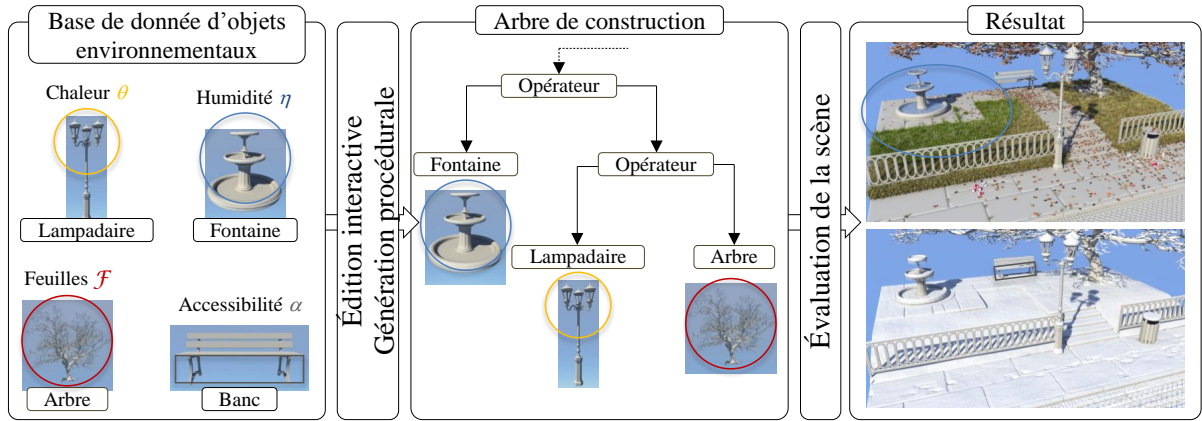


FIGURE 2.3 – Processus de création d'une scène décorée à partir d'objets environnementaux.

La scène décorée est construite en traversant l'arbre de construction. La scène finale est obtenue en combinant la géométrie de base des objets environnementaux avec les décorations géométriques produites par les différents effets. Trois types de requêtes sont nécessaires à l'évaluation de l'arbre de construction : obtention de la géométrie des objets R_G , évaluation de l'environnement R_e , et génération des décorations géométriques par les effets $R_{\mathcal{E}}$.

Géométrie La requête d'obtention de la géométrie des objets R_G est simple : elle consiste en une union de la géométrie des différents objets environnementaux de la scène, puisque la géométrie des objets en eux-mêmes ne dépend pas de l'environnement. Cette requête, triviale dans le cadre d'un objet environnemental simple, est plus complexe dans le cadre d'un objet environnemental paramétré (Section 2.1.2).

Environnement La requête d'évaluation de l'environnement R_e quantifie les différents champs d'influence et détermine la valeur d'un effet ou d'un phénomène en un point donné. Elle est détaillée dans le chapitre 3.

Effets La requête de génération des décorations $R_{\mathcal{E}}$ crée les décorations géométriques et s'appuie sur la requête d'évaluation de l'environnement, afin de quantifier spatialement les différents champs d'influence. La génération des décoration géométriques est analysée dans le chapitre 4.

La scène peut être contrôlée à quatre niveaux différents : par objet environnemental, par groupe d'objets environnementaux, localement, et globalement. Le contrôle par objet environ-

nemental correspond à la définition du comportement d'un objet environnemental, c'est à dire à la façon dont il influence l'environnement (par exemple en dégageant de la chaleur). Grâce à la structure hiérarchique de l'arbre, il est possible de lier différents objets environnementaux par un opérateur de combinaison pour former un groupe, et de redéfinir ensuite le comportement de ce groupe afin qu'il ne soit pas identique à l'union des comportements des différents objets qui le composent. Ce type d'opération, détaillé dans la section ??, permet par exemple de remplacer des effets complexes par des effets plus simples et moins coûteux afin d'accélérer l'évaluation de la scène lorsque cela est possible (par exemple dans le cas où des décorations géométriques très détaillées seront très éloignées du point de vue). Le contrôle local correspond à l'utilisation de champs d'influence de taille limitée, qui définissent un effet ou un phénomène dont la valeur n'est pas constante à l'échelle de la scène. Ce sera par exemple le cas du champ modélisant la chaleur émanant d'une bouche de métro. À l'inverse, le contrôle global correspond à des champs d'influence englobant l'intégralité de la scène qui sont utilisés pour définir un effet ou un phénomène constant à l'échelle de la scène, comme par exemple la chute de neige.

2.1 Objets environnementaux

Un objet environnemental est défini comme un objet procédural, feuille de l'arbre de construction, qui se décore automatiquement en fonction des caractéristiques de l'environnement. C'est un objet géométrique étendu avec deux aspects : il peut avoir un impact sur l'environnement par le biais de ses champs d'influence, et il réagit à l'environnement en modifiant son apparence par le biais de ses effets (Figure 2.4). Formellement, un objet environnemental \mathcal{O} est constitué d'une géométrie \mathcal{G} , d'effets $\mathcal{E}(e)$, et de champs d'influence \mathcal{C} :

$$\mathcal{O} = (\mathcal{G}, \mathcal{E}(e), \mathcal{C})$$







Lampadaire					
Géométrie \mathcal{G}	Effets \mathcal{E}				Champs \mathcal{C}
	Neige \mathcal{N}	Glace $\mathcal{G}\ell$	Feuilles \mathcal{F}	Herbe \mathcal{H}	Température θ
					

FIGURE 2.4 – L'objet environnemental *lampadaire* est créé en liant des effets et des champs d'influence à un objet géométrique.

Sans perte de généralité, nous considérons l'objet géométrique \mathcal{G} comme un maillage polygonal. Il est cependant important de remarquer que \mathcal{G} n'est pas réduit à ce type d'objet et peut être obtenu par n'importe quelle technique, que ce soit par modélisation surfacique comme les surfaces implicites ou par des représentations volumiques.

2.1.1 Objet environnemental statique

Les objets environnementaux statiques sont des objets environnementaux dont la géométrie \mathcal{G} ne varie pas, par opposition aux objets environnementaux paramétrés (section 2.1.2). Le processus de construction d'un objet environnemental statique est décrit par la figure 2.5. Dans cet exemple, le modèle géométrique d'origine est une fontaine. Elle peut se couvrir de neige et de glace, et des feuilles peuvent se déposer dans les parties concaves comme les vasques : il faut

donc définir des effets de neige, de glace et de feuilles dans ces zones. Des effets de neige sont fixés sur les surfaces où la neige peut se déposer. Des effets de glace sont ajoutés dans les zones d'écoulement d'eau, puisque des stalactites sont susceptibles de s'y former. Des feuilles mortes peuvent se déposer au fond des vasques, un effet de feuille est donc joint à chacune d'elles. Un champ d'humidité est ajouté à l'intérieur de chaque vasque pour exprimer le rôle de l'eau. Finalement, des champs d'accessibilité sont ajoutés pour diminuer la hauteur de neige là où la géométrie est moins accessible. Cela permettra de définir une quantité de neige plus faible dans ces régions.







Fontaine						
Géométrie \mathcal{G}	Effets \mathcal{E}				Champs \mathcal{C}	
	Neige \mathcal{N}	Glace $\mathcal{G}\mathcal{I}$	Feuilles \mathcal{F}	Herbe \mathcal{H}	Humidité \mathcal{H}	Accessibilité \mathcal{A}
				\emptyset		

FIGURE 2.5 – Extension de la géométrie de la fontaine par des effets de neige, de glace et de feuilles, et des champs d'humidité et d'accessibilité.

Un exemple de réaction de l'objet environnemental *fontaine* est montré dans la figure 2.6. Un champ de neige \mathcal{N}_1 l'englobe presque entièrement, et des champs de glace \mathcal{I}_1 et \mathcal{I}_2 ont été ajoutés au niveau des vasques. Lorsque la fontaine réagit à l'environnement, des stalactites se forment sous les vasques, dans l'espace situé à l'intérieur des deux champs de glace. La fontaine se couvre également de neige, exception faite de la zone qui n'est pas à l'intérieur du champ correspondant.

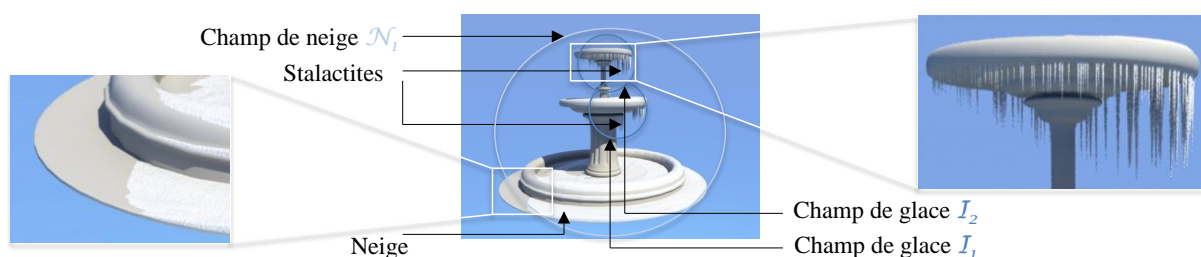


FIGURE 2.6 – Réaction de l'objet environnemental *fontaine* à un champ de neige et des champs de glace.

2.1.2 Objets environnementaux paramétrés

Cette seconde catégorie d'objets environnementaux désigne des objets qui, en plus d'être réactifs à l'environnement, sont capables de s'adapter automatiquement à des paramètres supplémentaires. La géométrie $\mathcal{G}(\mathcal{P})$ des objets environnementaux paramétrés peut soit être composée d'une seule composante connexe, soit être une union de différentes sous-parties $\mathcal{G}_i(\mathcal{P})$. Nous décrirons le fonctionnement des objets environnementaux paramétrés dans cette section avec un exemple simple. Un cas d'objet environnemental paramétré complexe, une fenêtre constituée de plusieurs sous-parties, est analysé en détails dans l'annexe A.

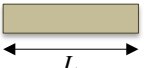
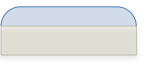



Boite					
Paramètres \mathcal{P}	Géométrie $\mathcal{G}(\mathcal{P})$	Effets $\mathcal{E}(\mathcal{P})$			
l		Neige $\mathcal{N}(\mathcal{P})$	Glace $\mathcal{G}(\mathcal{P})$	Feuilles $\mathcal{F}(\mathcal{P})$	Herbe $\mathcal{H}(\mathcal{P})$
L					

 FIGURE 2.7 – L’objet environnemental paramétré *boite*.

L’exemple montré dans la figure 2.7, l’objet environnemental *boite*, est un parallélépipède rectangle dont la hauteur est fixe et dont la largeur et la longueur sont respectivement L et l . Comme il est vu de face, l n’est pas visible sur la figure. Les effets Neige, Feuille et Herbe sont placés sur sa face horizontale supérieure. Cet objet environnemental paramétré ne produit par contre pas de stalactites de glace.

Classiquement, utiliser une opération de transformation sur un modèle géométrique consiste à appliquer la matrice associée à cette transformation sur le modèle, dont les différents sommets sont déplacés en fonction de cette transformation. Dans le contexte de la création de décorations géométriques, il est impossible d’utiliser ce mode de transformation, puisque les détails générés seraient également déformés lors de l’opération. La cohérence spatiale entre les détails générés et l’environnement serait également perdue. Considérons l’objet environnemental *boîte* plongé dans un champ de feuilles. L’image de gauche de la figure 2.8 montre la boîte dans sa configuration d’origine, avec $L = 1$ et $l = 1$. L’image centrale montre la même boîte transformée par une homothétie de vecteur $v_h(2, 1, 1)$, après la génération des décorations géométriques. La géométrie de la boîte est transformée par cette homothétie, et les décorations géométriques, les feuilles, sont également mises à l’échelle. Elles perdent donc leur apparence d’origine.

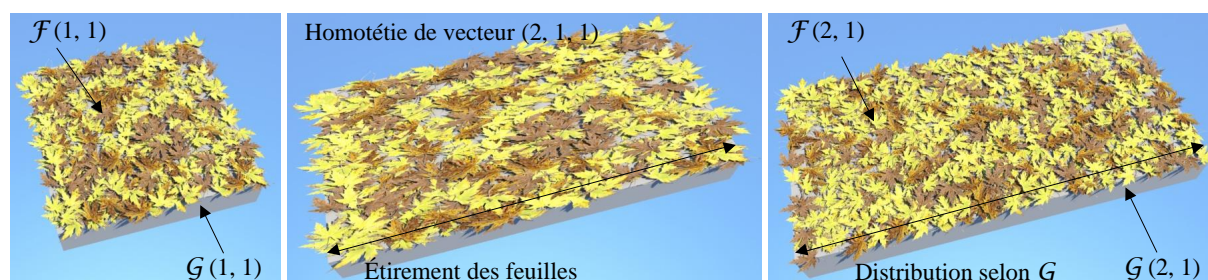


FIGURE 2.8 – Différence entre un redimensionnement par homothétie et une distribution adaptée à la géométrie de l’objet environnemental paramétré.

Au lieu de transformer les détails produits par les effets, notre système adapte automatiquement, lors de la spécification des paramètres, les dimensions des effets. Cette phase a lieu lors du placement de l’objet environnemental dans la scène, c’est à dire en amont de la requête de génération des décorations géométriques $R_{\mathcal{E}}$. L’image de droite montre une configuration différente de la boîte, avec $L = 2$ et $l = 1$. Les paramètres de cette nouvelle configuration ont été choisis pour que \mathcal{G} soit identique à celle de la première configuration après la transformation par homothétie. Lors de la requête de génération des décorations $R_{\mathcal{E}}$, comme les effets ont été créés avec des dimensions correspondant à celles de la boîte, les décorations géométriques vont être générées de façon cohérente. Les feuilles recouvrent l’intégralité de la surface associée à l’effet

de feuilles, mais cette fois, elles ne sont pas déformées : c'est leur distribution qui a été adaptée.

En redimensionnant les effets en fonction des paramètres des objets au lieu de redimensionner les décorations géométriques produites par les effets, nous nous assurons d'obtenir des décorations géométriques adaptées à leur support, la géométrie des objets environnementaux. Quelles que soient les dimensions de la géométrie des objets environnementaux, la précision des décorations géométriques créées lors de l'évaluation des effets (par exemple la précision du maillage de la surface de neige) est identique, ce qui permet d'utiliser conjointement différentes configurations d'un même objet environnemental paramétré ou différents objets environnementaux.

2.2 Opérateurs de combinaison

Les nœuds de l'arbre de construction sont des opérateurs de combinaison qui structurent la scène et créent une hiérarchie de volumes englobants accélérant les différentes requêtes. Ils combinent les différentes feuilles de l'arbre pour produire la scène finale. Considérons une scène simple, composée de deux objets environnementaux et d'un champ d'influence. L'arbre de construction de cette scène et le résultat qu'il produit sont présentés dans la figure 2.9.

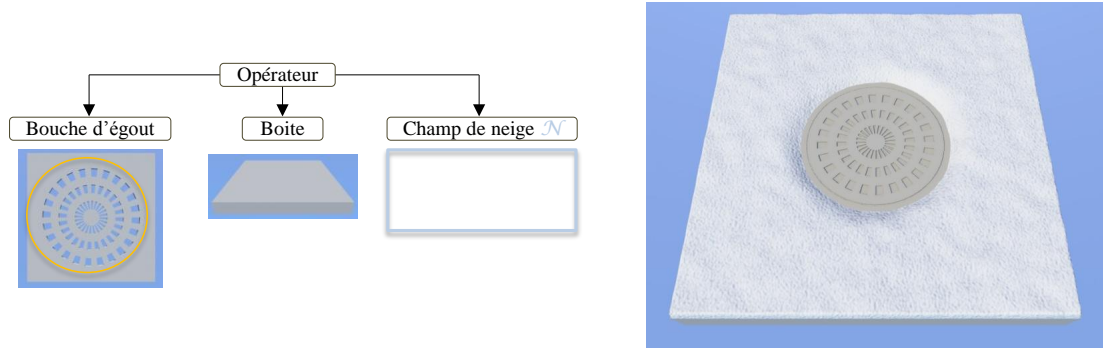


FIGURE 2.9 – Arbre de construction et résultat produit par une scène simple, assemblée à partir des objets environnementaux *bouche d'égout* et *boite*, et d'un champ de neige.

La géométrie de base de la scène \mathcal{G} , qui ne comporte aucune décoration, est la combinaison des géométries de base \mathcal{G}_i des n objets environnementaux de la scène. Elle est assemblée en réalisant un parcours complet de l'arbre et en réalisant une union des \mathcal{G}_i :

$$\mathcal{G} = \bigcup_{i=1}^n \mathcal{G}_i$$

Les champs d'influence sont combinés par les nœuds de l'arbre par une fonction f :

$$\mathcal{C} = \bigcup_{i=1}^n f \mathcal{C}_i$$

Les fonctions de combinaison varient selon le type de champ et sont détaillées dans le chapitre 3. Un second type de nœuds ayant un comportement spécifique peut-être utilisé pour mettre en place des accélérations du parcours de l'arbre lors de l'évaluation de l'environnement. Le fonctionnement de ce type de nœud est également détaillé dans le chapitre 3.

Au niveau d'un objet environnemental \mathcal{O} comportant m effets, les décorations géométriques de \mathcal{O} sont l'union des décorations géométriques générées par les différents effets de cet objet :

$$\mathcal{E}_{\mathcal{O}}(e) = \bigcup_{i=1}^m \mathcal{E}_i(e)$$

Les décorations géométriques d'une scène sont la combinaison des décorations géométriques des objets environnementaux de la scène :

$$\mathcal{E}(e) = \bigcup_{i=1}^n \mathcal{E}_{\mathcal{O}_i}(e)$$

La géométrie de la scène finale, c'est à dire l'ensemble de la géométrie de base de la scène et des décorations géométriques des n objet environnemental est définie par l'union $\mathcal{G} \cup \mathcal{E}(e)$.

3 Évaluation des effets

Afin de comprendre le fonctionnement des objets environnementaux, il est nécessaire d'analyser comment ils réagissent à l'environnement. Cette section détaille le principe du mécanisme de génération des effets en fonction de l'environnement au niveau des objets environnementaux, sans détailler les mécaniques internes des champs d'influence ni des effets : l'environnement est décrit en détail dans le chapitre 3, et la production des décorations géométriques par les effets dans le chapitre 4.

3.1 Réaction à l'environnement

Dans cette partie, nous analysons le mécanisme de la génération des décorations géométriques par les effets. Au début de ce chapitre, nous avons décrit quatre façons de contrôler une scène. Dans l'exemple suivant, nous illustrerons le contrôle *global* et le contrôle *local*, respectivement par l'utilisation d'un champ de neige qui définit la quantité de neige sur l'intégralité de la scène, et par l'utilisation de deux champs de température pour diminuer localement la hauteur de neige.

Considérons une scène simple, constituée d'un unique objet environnemental \mathcal{O} que l'on peut voir dans la figure 2.10, doté de deux effets de neige \mathcal{E}_{N_1} et \mathcal{E}_{N_2} . À ce stade, la scène ne contient aucun champ d'influence : l'environnement est $e = \emptyset$. \mathcal{O} ne produit donc pas de décorations géométriques.

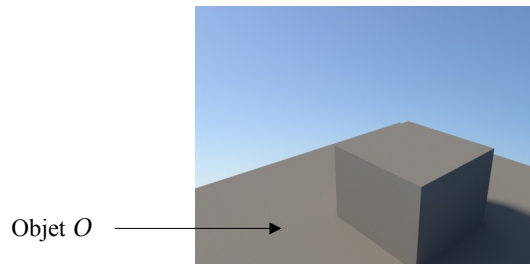


FIGURE 2.10 – Une scène simple composée d'un objet environnemental \mathcal{O} .

Plongeons maintenant la scène dans un champ de neige \mathcal{N} (Figure 2.11). Il s'agit d'un exemple de contrôle global, puisque ce champ influence de manière identique l'intégralité de la scène. L'environnement est donc maintenant constitué d'un unique champ d'influence : $e = \{\mathcal{N}\}$. \mathcal{O} réagit à ce champ et ses effet de neige génèrent les couches de neige $\mathcal{E}_{N_1} \rightarrow \mathcal{N}(e)$ et $\mathcal{E}_{N_2} \rightarrow \mathcal{N}(e)$.

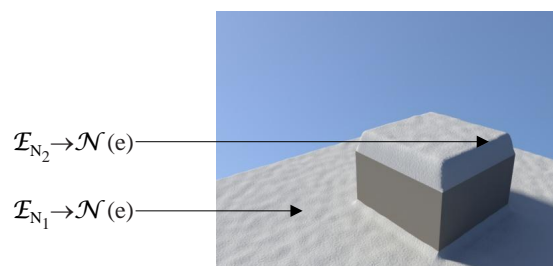


FIGURE 2.11 – La scène est plongée dans un champ de neige \mathcal{N} .

L'introduction d'un champ de température θ_1 dans la scène est un exemple de contrôle local : ce champ est placé pour représenter un phénomène dont la zone d'influence réduite affectera la scène localement (Figure 2.12). L'environnement devient donc $e = \{\mathcal{N}, \theta_1\}$. La réaction de \mathcal{O} va donc être différente, puisque le champ de température va réduire la quantité de neige qui s'est déposée dans son aire d'effet. On choisit que la chaleur générée par θ_1 soit plus forte à proximité de son centre : $\mathcal{E}_{N_1} \rightarrow \mathcal{N}(e)$ varie peu par rapport à l'itération précédente, alors que $\mathcal{E}_{N_2} \rightarrow \mathcal{N}(e)$ est beaucoup plus affectée par le champ de température, au point qu'une partie de la face supérieure de \mathcal{O} ne soit plus couverte de neige.

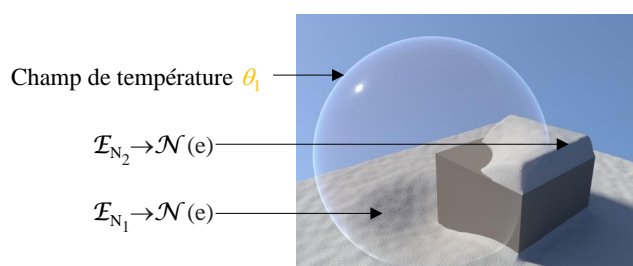


FIGURE 2.12 – Un champ de température θ_1 vient modifier la réaction de l'objet environnemental \mathcal{O} .

Un second champ de température θ_2 est ajouté à la scène, et l'environnement devient $e = \{\mathcal{N}, \theta_1, \theta_2\}$ (Figure 2.13). La couche de neige générée par \mathcal{E}_{N_2} est donc différente par rapport à la scène précédente, puisque cet effet se situe dans l'aire d'effet des deux champs : la face supérieure de \mathcal{O} reçoit moins de neige que quand il n'était en présence que de θ_1 . À l'inverse, comme il n'existe pas d'intersection entre \mathcal{E}_{N_1} et θ_2 , \mathcal{E}_{N_1} n'est pas influencée par θ_2 . $\mathcal{E}_{N_1} \rightarrow \mathcal{N}(e)$ est donc identique par rapport à l'étape précédente.

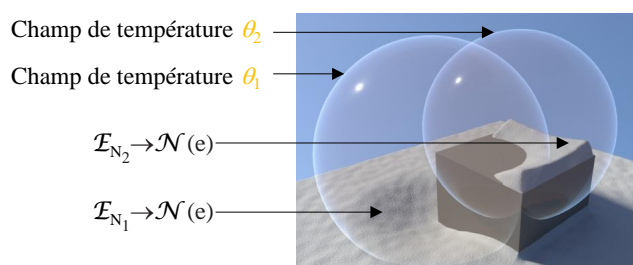


FIGURE 2.13 – L'influence du champ de température θ_2 se combine à celle de θ_1 .

3.2 Interactions

Le second aspect à analyser est l'inter-influence entre les objets environnementaux, qui peuvent s'influencer entre eux par le biais de leurs champs d'influence. Lorsque l'on introduit, déplace ou supprime un objet environnemental qui contient des champs d'influence dans une scène, ces champs sont déplacés avec l'objet. Il est ainsi possible de faire varier l'apparence d'une scène uniquement en manipulant les objets environnementaux qui la composent : il s'agit du contrôle par objet.

Pour analyser l'inter-influence des objets environnementaux, considérons d'abord l'objet environnemental *arrêt de bus* \mathcal{O}_1 qui contient un champ d'accessibilité α_1 . Dans la figure 2.14, il est posé sur le sol, et la scène est englobée dans un champ de neige. L'arrêt de bus se couvre de neige, et son champ d'accessibilité empêche la neige de se déposer sur le sol.

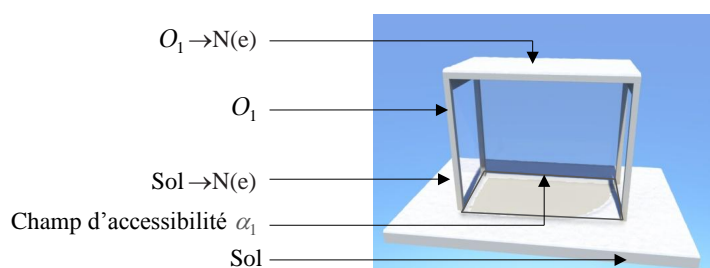


FIGURE 2.14 – L'objet environnemental *arrêt de bus* est doté d'un champ d'accessibilité.

Le second objet environnemental utilisé dans cet exemple est le *banc* \mathcal{O}_2 qui est similaire à l'arrêt de bus : il contient également un champ d'accessibilité, noté α_2 . Lorsqu'il est posé au sol et que la scène est plongée dans un champ de neige (Figure 2.15), il se couvre de neige et diminue également la quantité de neige qui se dépose sur le sol.

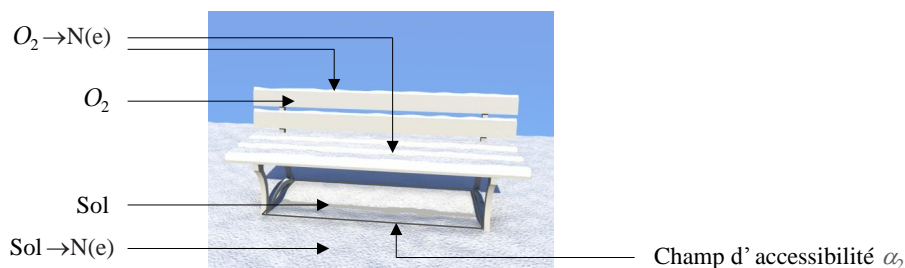


FIGURE 2.15 – L'objet environnemental *banc* comporte également un champ d'accessibilité.

Réunissons maintenant \mathcal{O}_1 et \mathcal{O}_2 dans une même scène, englobée par un champ de neige (Figure 2.16). Les deux objets environnementaux ne sont pas en contact, il n'y a donc pas d'interactions entre eux. Ils se couvrent tous les deux de neige.

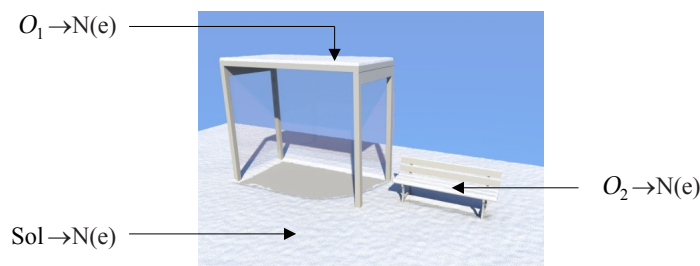


FIGURE 2.16 – Réunion des objets environnementaux \mathcal{O}_1 et \mathcal{O}_2 dans une scène.

Dans la figure 2.17, \mathcal{O}_2 est partiellement couvert par \mathcal{O}_1 . Une partie de \mathcal{O}_2 est donc située à l'intérieur du champ d'accessibilité α_1 de \mathcal{O}_1 . Lorsque les effets de \mathcal{O}_2 créent les décorations géométriques en réagissant à l'environnement, ils sont influencés par α_1 . En conséquence, seule la partie de \mathcal{O}_2 située à l'extérieur de \mathcal{O}_1 se couvrira de neige, puisque sur la partie de \mathcal{O}_2 située à l'intérieur de \mathcal{O}_1 , α_1 ramène la quantité de neige à 0.

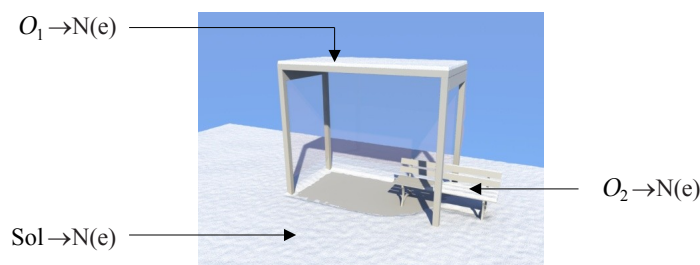


FIGURE 2.17 – \mathcal{O}_1 recouvre partiellement \mathcal{O}_2 .

\mathcal{O}_1 est à nouveau déplacé en direction de \mathcal{O}_2 et le recouvre a présent totalement (Figure 2.18). Les effets de \mathcal{O}_2 sont maintenant intégralement enveloppés dans α_1 . Lors de la requête de génération des décorations géométriques, la hauteur de neige sera 0 sur la totalité de la surface de \mathcal{O}_2 , et les effets de \mathcal{O}_2 ne généreront donc pas de neige.

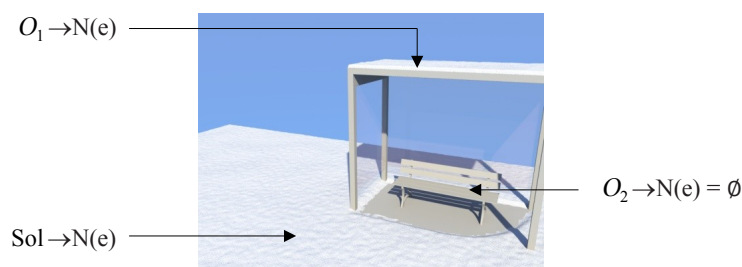


FIGURE 2.18 – \mathcal{O}_1 recouvre entièrement \mathcal{O}_2 .

Nous illustrons ici le fait que les champs d'influence qui font partie d'un objet environnemental contribuent de la même façon à l'environnement que les champs d'influence de l'environnement qui ne sont liés à aucun objet. Lors de la requête environnementale, aucune différence n'est faite entre les champs de l'environnement et ceux des objets environnementaux. Ainsi, il est possible de faire varier l'apparence d'une scène uniquement en manipulant les objets environnementaux qui la composent.

4 Création des scènes

L'obtention des scènes décorées se fait en trois étapes (Figure 2.19). La première étape est la création des objets environnementaux à partir des objets géométriques. La seconde étape est la génération de l'arbre de construction de la scène organisant les différents objets environnementaux et champs d'influence. La scène finale est obtenue automatiquement à partir de l'arbre de construction, en combinant la géométrie des objets et les détails géométriques créés par les effets.

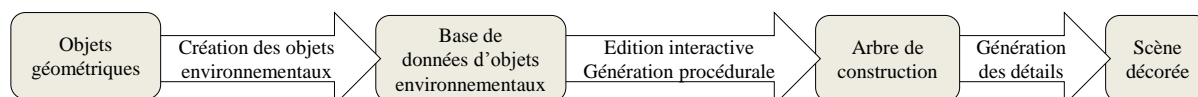


FIGURE 2.19 – Flux de création d'une scène dans notre système.

4.1 Base de données d'objets environnementaux

La première étape nécessaire à l'obtention d'une scène décorée consiste à créer une base de données d'objets environnementaux (Figure 2.20) à partir d'objets géométriques, dans laquelle on différencie la catégorie des objets environnementaux statiques (arbre, statue, etc.) de celle des objets environnementaux paramétrés (fenêtre, bâtiment, etc.). Cette étape est réalisée en liant des champs d'influence et des effets à un objet géométrique (section 2.1). Deux méthodes existent pour créer les objets environnementaux : soit par édition interactive, soit par génération procédurale. La première est employée pour les objets environnementaux statiques et consiste à décrire explicitement l'ensemble des effets et des champs d'influence attachés à un objet environnemental. La seconde méthode est utilisée dans le contexte des objets environnementaux paramétrés. En fonction des paramètres d'un objet environnemental, elle ajoute et configure automatiquement ses différents effets et ses différents champs d'influence.

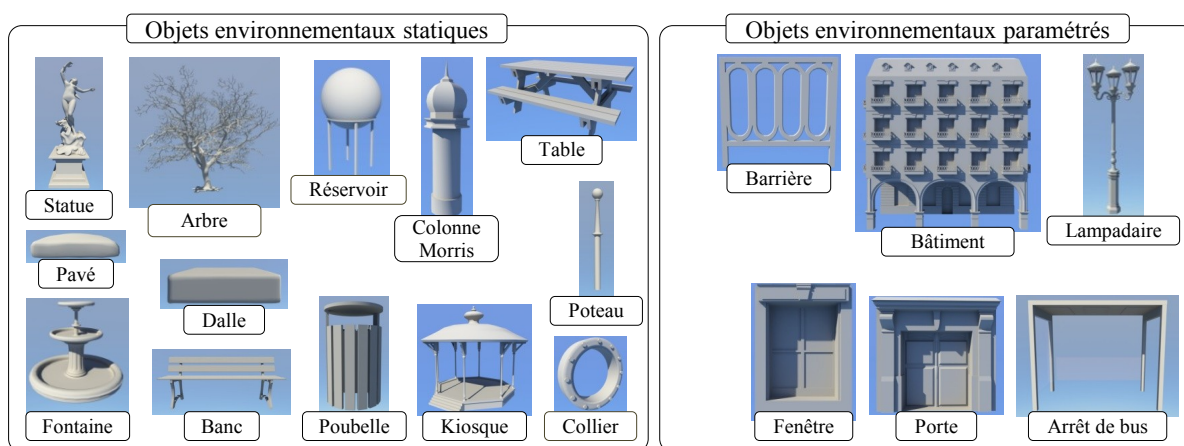


FIGURE 2.20 – Base de donnée des objets environnementaux construits dans le cadre de cette thèse.

Chaque objet environnemental peut ensuite être instancié et réutilisé autant de fois que nécessaire : la phase de création n'a lieu qu'une fois, avant d'ajouter l'objet à la base de données, et non à chaque fois que l'objet est placé dans une scène. En conséquence, si une modification est faite sur un objet environnemental (tel que l'ajout ou le retrait d'un champ d'influence),

cette modification se répercutera sur l'ensemble des instances de cet objet.

4.2 Création de l'arbre de construction

La seconde étape est la création de l'arbre de construction de la scène (Figure 2.21). Dans notre système, il peut être créé soit par édition interactive, soit par génération procédurale.

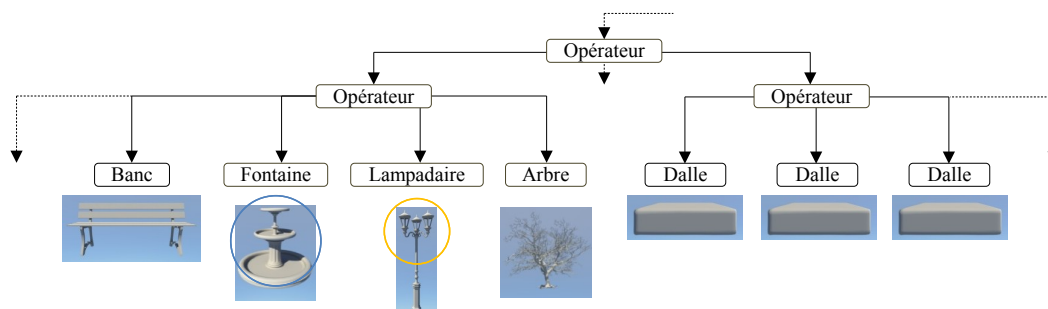


FIGURE 2.21 – Sous partie de l'arbre de construction correspondant aux scènes de la figure 2.26.

4.2.1 Édition interactive

L'édition interactive permet d'obtenir une visualisation quasi instantanée de la scène décorée, tout en ajoutant, déplaçant ou supprimant des objets environnementaux et des champs d'influence. Elle est particulièrement adaptée à la création de petites scènes, ou à l'amélioration de scènes existantes dans lesquelles l'utilisateur veut insister sur le réglage précis d'effets particuliers.

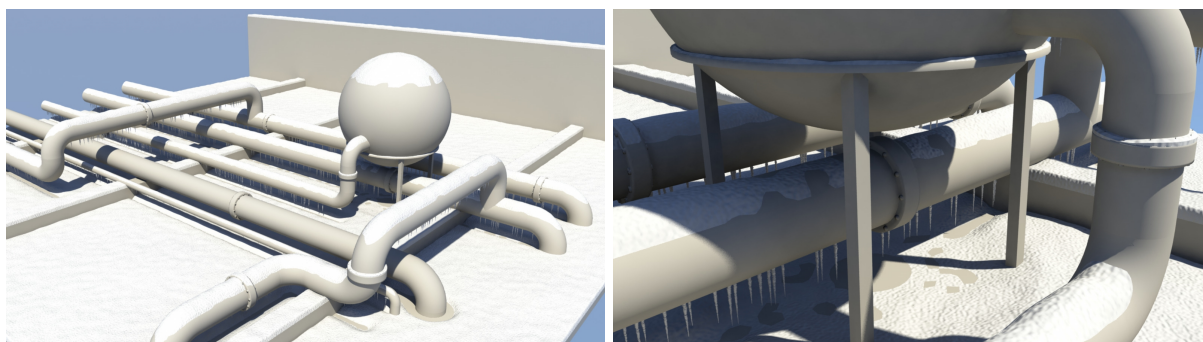


FIGURE 2.22 – Une raffinerie en hiver, couverte de neige et de stalactites.

La raffinerie présentée dans la figure 2.22 est un exemple type de scène créée par édition interactive. Cette scène de petite taille n'est pas particulièrement structurée, comme on peut le voir par le biais de son arbre de construction (Figure 2.23). L'édition interactive convient parfaitement à l'édition de la taille des stalactites ou de la quantité de neige qui fond à cause des tuyaux dégageant de la chaleur.

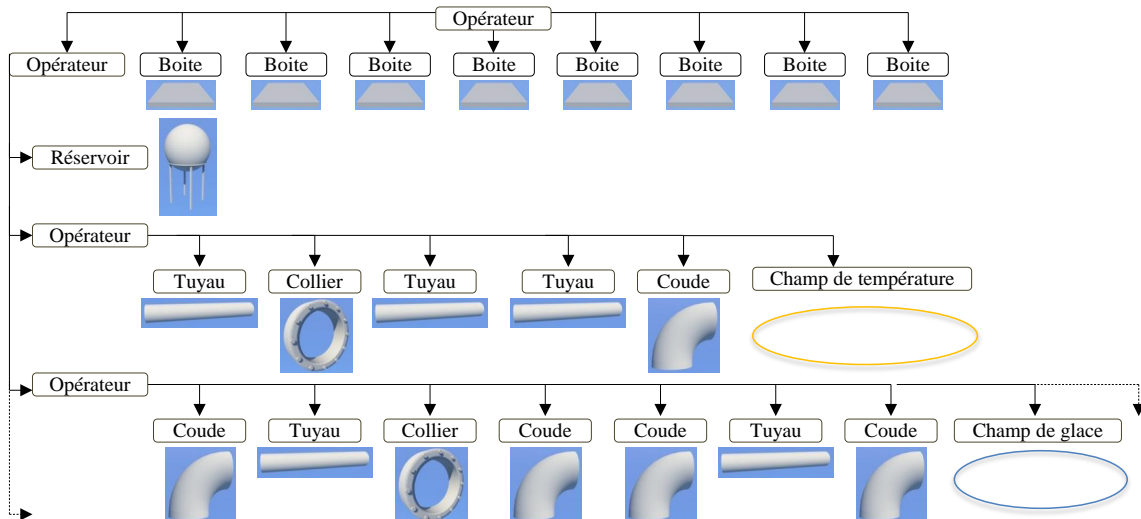


FIGURE 2.23 – Sous-partie de l’arbre de construction de la raffinerie.

4.2.2 Génération procédurale

Contrairement à l’édition interactive, la génération procédurale est utilisée pour construire automatiquement de grandes scènes, pour créer des structures à taille variable que l’on peut facilement exprimer avec des grammaires de formes, ou pour créer des objets environnementaux paramétrés. Nous l’utilisons par exemple pour produire facilement et rapidement différentes déclinaisons d’un même bâtiment. La figure 2.24 montre une sous partie de l’arbre de construction d’un des bâtiments paramétrés que l’on peut voir dans la figure 2.25. Comme ce type de bâtiment est très structuré, la génération procédurale est bien adaptée à la création de son arbre de construction : le découpage du bâtiment en différentes composantes (façades, toit, etc.) est intuitif, et ce type de contenu peut être généré facilement avec des grammaire de formes.

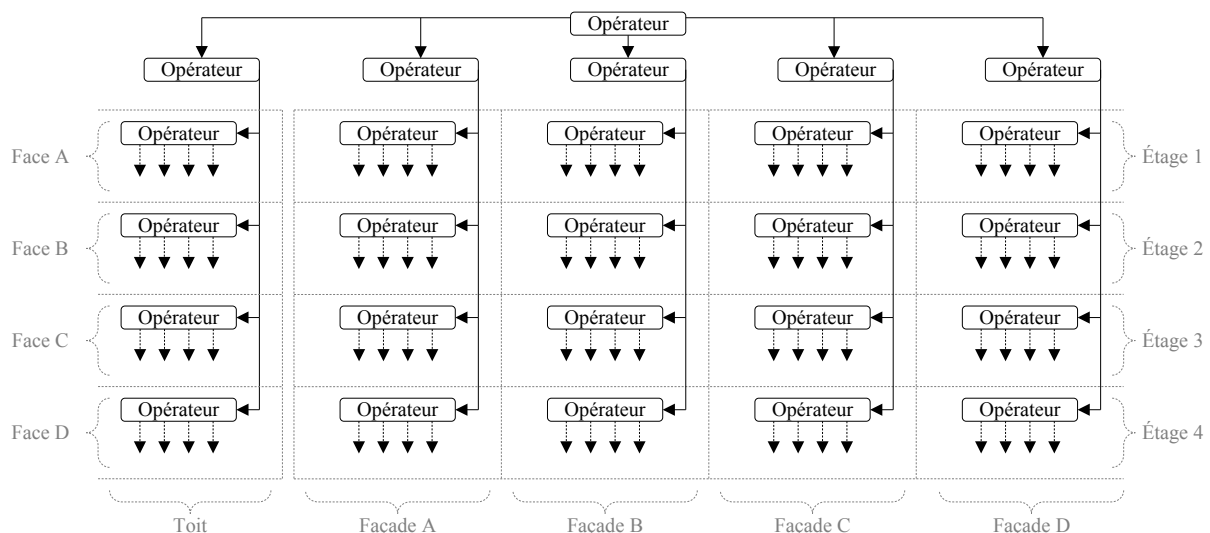


FIGURE 2.24 – Sous-partie de l’arbre de construction d’un bâtiment procédural.

La figure 2.25 montre un exemple de scène où différentes structures ont été créées procéduralement. Les immeubles qui entourent le parc sont générés automatiquement d’après une empreinte au sol, et il peuvent varier selon un grand nombre de critères : nombre d’étages, type

de façade, taille des fenêtres, etc. La barrière est un objet environnemental paramétré composée de cinq sous parties. Elle s'adapte automatiquement à différentes contraintes de dimension (hauteur des boucles, hauteur totale, etc.)



FIGURE 2.25 – Les bâtiments et la barrière entourant ce parc sont générés procéduralement.

4.3 Génération de la scène décorée finale

Les détails géométriques de la scène sont calculés lors de l'évaluation de l'arbre, lorsque les différents effets des objets environnementaux réagissent à l'environnement, c'est à dire lors de l'exécution de la requête de génération des décorations géométriques R_E (Section 2). L'importance de l'environnement dans cette phase est mise en avant dans la figure 2.26 : on obtient, en faisant varier uniquement l'environnement, des résultats totalement différents. L'image de gauche a été produite à l'aide d'un champ de neige qui englobe toute la scène et de champs de glace situés au niveau de la fontaine. Dans l'image de droite, ce sont un champ d'herbe et le champ d'humidité de la fontaine qui provoquent la génération de l'herbe, et le champ de feuilles de l'arbre qui déclenche l'apparition des feuilles mortes.



FIGURE 2.26 – Production de décorations différentes à partir de l'arbre de construction de la figure 2.21.

5 Conclusion

Dans ce chapitre, nous avons présenté un système pour construire des scènes dont l'apparence varie en fonction de l'environnement. Ce système repose sur les objets environnementaux, des objets géométriques procéduraux augmentés par des champs d'influence et des effets. Les champs d'influence sont des champs scalaires en trois dimensions qui contrôlent les effets et phénomènes de la scène. L'ensemble des champs d'influence forme l'environnement qui quantifie des informations, telles que la quantité de neige ou la présence de feuilles, en tout point de l'espace.

Les effets réagissent à l'environnement et génèrent les décorations géométriques correspondantes.

Les scènes sont organisées avec un arbre de construction. Les feuilles de cet arbre sont des objets environnementaux, et ses nœuds sont des opérateurs de combinaisons. L'utilisation de cette structure rend aisée la génération de scènes aussi bien par des systèmes procéduraux (e. g. grammaires de forme ou *L-system*) que par édition interactive. Cette structure arborescente apporte également la possibilité de redéfinir le comportement de chaque sous-partie de l'arbre au niveau des opérateurs de combinaison, afin par exemple d'accélérer l'évaluation de scènes très complexes ou de gérer le niveau de détail des décorations géométriques.

La scène finale décorée est construite à partir de l'arbre de construction grâce à trois requêtes : géométrie des objets, évaluation de l'environnement, et génération des décorations géométriques. La requête de géométrie des objets R_G unit la géométrie des différents objets de l'arbre. La requête d'évaluation de l'environnement R_e quantifie spatialement l'ensemble des champs d'influence. Elle est très largement utilisée par la requête de génération des décorations $R_{\mathcal{E}}$ qui crée les détails géométriques. Le contrôle des scènes peut être effectué à quatre niveaux différents. Par objet en ajoutant des objets environnementaux qui comportent des champs d'influence. Par groupe d'objets en redéfinissant le comportement d'un opérateur de combinaison de l'arbre. Localement avec un champ d'influence à la portée limitée, et globalement avec un champ d'influence qui englobe l'intégralité de la scène.

Dans le chapitre suivant, nous analysons en détail le fonctionnement de l'environnement et des champs d'influence et nous détaillons comment les informations nécessaires aux différentes requêtes sont quantifiées.

Chapitre 3

Environnement

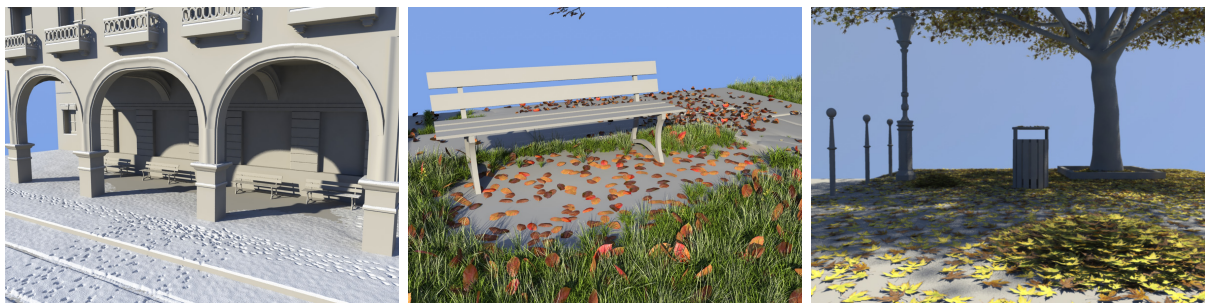


FIGURE 3.1 – De gauche à droite : Champs d’accessibilités créant des traces de véhicules et de piétons, champs d’accessibilité modélisant la diminution de l’herbe et champs de feuilles créant des tas.

Table des matières

1	Introduction	57
2	Définition de l’environnement	57
3	Les champs d’influence	58
3.1	Spécificités des champs d’herbe et de feuilles	59
3.2	Primitives à squelette	59
3.3	Primitives à base d’image	62
4	Combinaison des champs d’influence	62
4.1	Cas général	63
4.2	Cas particulier de l’accessibilité	63
5	Accélération	66
6	Conclusion	68

1 Introduction

Dans notre système, l'environnement désigne un concept récurrent en informatique graphique. C'est un ensemble de facteurs variés, qui ont une influence sur l'apparence de l'objet étudié. Parmi ces facteurs on trouve par exemple, et de façon non exhaustive, l'exposition à la lumière qui guide la croissance des plantes par le biais de l'héliotropisme [30], les différents types de ressources contenues dans les sols (eau, minéraux, etc.) nécessaires à la croissance des organismes végétaux [4], [80], [38], la composition de l'atmosphère et des précipitations qui influent sur la dégradation des matériaux de construction [82], la présence d'obstacles physiques qui jouent un rôle sur la distribution et la croissance des végétaux [56], [111], ou même des contraintes artificielles et arbitraires mises en places pour permettre à l'utilisateur de piloter facilement un processus de génération [115], [8].

Plusieurs formalismes existants ont été adaptés afin de prendre en compte les données de l'environnement. C'est par exemple le cas des *L-system* paramétrés, une version étendue des *L-system* qui permet aux différents symboles de communiquer entre eux [59]. Le modèle de grammaire de forme introduit dans [91] est un autre exemple de système produisant des résultats nettement améliorés grâce à ses possibilités de communication.

Il serait possible de calculer les effets et phénomènes par différentes simulations. Le chapitre 1 référence différentes techniques de simulation appliquées à la modélisation de neige, à la formation de glace, à la distribution de feuilles ou encore aux transferts de chaleur. Cependant, ces méthodes sont coûteuses, y compris quand elles tirent parti des capacités de calcul parallèle des processeurs récents, et elles sont en général limitées à des situations spécifiques.

Afin de ne pas être sujet aux différents inconvénients des simulations, nous avons choisi d'utiliser une autre approche, la modélisation par champ d'influence. Elle fournit à l'utilisateur un outil intuitif et efficace, qui permet un contrôle précis pour l'édition de scènes complexes. De plus, elle est plus rapide de plusieurs ordres de grandeur que les méthodes de simulation. Parmi les méthodes auxquelles nous pouvons nous comparer, les méthodes de neige sont les plus pertinentes : ce sont celles qui ont été les plus étudiées, et ce à travers des approches très différentes. Les méthodes de neige présentées dans le chapitre 1 permettent de calculer de petites scènes en quelques secondes, et les grandes scènes nécessitent des temps de calculs allant de plusieurs secondes à plusieurs heures. Notre système permet la génération de petites scènes de neige en quelques dixièmes de secondes, et en quelques secondes seulement pour les grandes scènes (Tableau 4.1).

Dans ce chapitre, nous proposons un nouveau formalisme pour décrire l'environnement. Il repose sur la combinaison de champs scalaires paramétrés, proches du formalisme des surfaces implicites. Notre but est de renforcer les possibilités de contrôle offertes aux utilisateurs, en leur fournissant un outil intuitif et efficace ; tout en étant rapide à évaluer. Nous unifions dans notre définition de l'environnement les informations nécessaires à différents processus de génération de décorations (neige, glace, herbe, feuilles) avec les informations qui perturbent ces processus de génération (accessibilité, chaleur, humidité).

2 Définition de l'environnement

L'environnement contient l'ensemble de caractéristiques et propriétés météorologiques de la scène. Il pilote les changements d'apparence en définissant les valeurs nécessaires aux différentes requêtes. De manière formelle, l'environnement est une fonction $e : \mathbb{R}^3 \rightarrow \mathbb{R}^n$ qui attribue à

chaque point \mathbf{p} de l'espace un ensemble de valeurs $e(\mathbf{p})$:

$$e(\mathbf{p}) = \left(\underbrace{\mathcal{N}(\mathbf{p}), \mathcal{G}l(\mathbf{p}), \mathcal{H}(\mathbf{p}), \mathcal{F}(\mathbf{p})}_{\mathcal{E}}, \underbrace{\alpha(\mathbf{p}), \eta(\mathbf{p}), \theta(\mathbf{p})}_{\mathcal{M}} \right)$$

Les fonctions désignées par le symbole \mathcal{E} sont les fonctions d'*effets*, chaque effet correspondant à un type de décoration. Dans notre plateforme, ce sont la neige, la glace, l'herbe et les feuilles. Les quantifications de ces effets en \mathbf{p} sont respectivement $\mathcal{N}(\mathbf{p})$, $\mathcal{G}l(\mathbf{p})$, $\mathcal{H}(\mathbf{p})$ et $\mathcal{F}(\mathbf{p})$. L'environnement centralise également les informations des *modificateurs*, notés \mathcal{M} . Ce sont des phénomènes qui altèrent et perturbent la génération des effets. Les modificateurs que nous utilisons sont l'accessibilité, l'humidité et la température, désignés respectivement par $\alpha(\mathbf{p})$, $\eta(\mathbf{p})$, et $\theta(\mathbf{p})$.

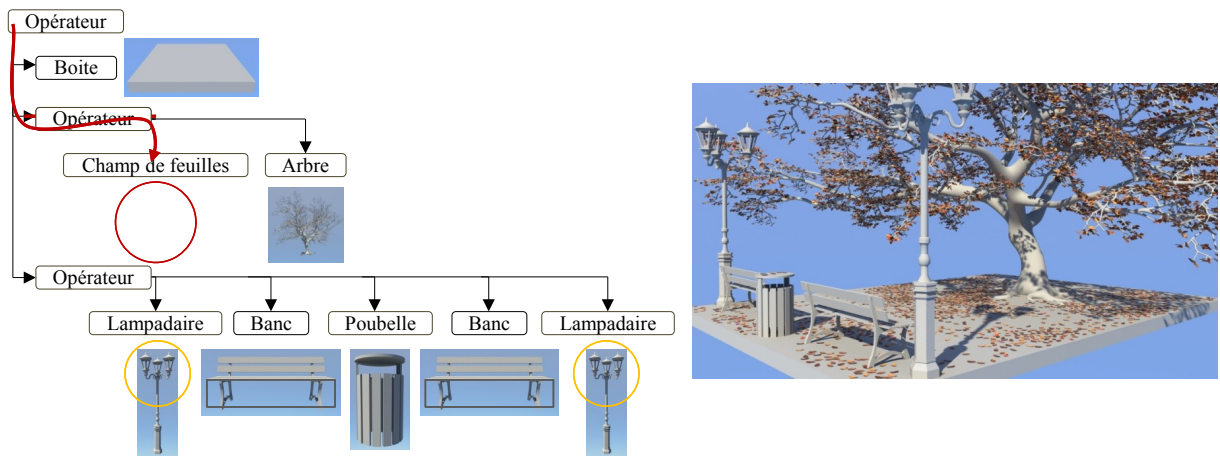


FIGURE 3.2 – Une scène automnale et son arbre de construction. La flèche rouge illustre le parcours de la requête de feuilles $\mathcal{F}(\mathbf{p})$.

L'évaluation de l'environnement est calculée récursivement, en traversant l'arbre de construction et en combinant l'intégralité des contributions des champs, qu'il s'agisse des champs liés aux objets environnementaux ou des champs de contrôle (Figure 3.2). Par conséquent, e est défini comme l'évaluation des champs \mathcal{C} au niveau du nœud racine de l'arbre de construction de la scène. Lors de l'évaluation de e , des branches de l'arbre peuvent être ignorées si elles ne comportent aucun champ d'influence.

3 Les champs d'influence

Le concept des champs d'influence correspond à la définition de la fonction de potentiel des surfaces implicites telles qu'elles sont utilisées dans [146]. Un champ d'influence est une fonction $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ qui associe une valeur à tout point \mathbf{p} de l'espace. Un champ d'influence est défini sur un squelette s , qui peut être virtuellement n'importe quel objet géométrique, à partir du moment où il existe une fonction $d(\mathbf{p})$ qui calcule la distance entre s et \mathbf{p} . Afin de pouvoir contrôler facilement et précisément l'action d'un champ d'influence, la valeur renvoyée par f ne doit pas être linéaire par rapport à $d(\mathbf{p})$, et les dérivées successives $f'(d)$ et $f''(d)$ doivent être nulles. Formellement, la fonction f est définie comme :

$$f(\mathbf{p}) = g \circ d(\mathbf{p})$$

ou d est la fonction de distance de \mathbf{p} à s et g est la fonction de distribution de potentiel. Différents types de fonctions ont été proposés pour la modélisation de surface implicite [146]. Dans notre système, nous utilisons la fonction de Wyvill standard $g(r) = (1 - (\frac{r}{R})^2)^3$ qui est de classe C^2 , R étant le rayon d'influence et r la distance de \mathbf{p} au squelette.

Nous utilisons les champs d'influence dans une grande diversité de scènes. Elles varient aussi bien dans leur nature, puisque l'on modélise des environnements urbains complexes ou des scènes naturelles, que dans leur taille, leur échelle allant du mètre à la centaine de mètres, ou leur niveau de détail, les scènes les moins précises comportant des détails qui mesurent une dizaine de centimètres alors que les plus précises comportent des détails inférieurs au centimètre. Il est donc nécessaire de disposer d'une grande variété de champs d'influence pour gérer toutes ces situations, et cette diversité passe par l'utilisation de squelettes hétérogènes.

3.1 Spécificités des champs d'herbe et de feuilles

En plus de leur fonction respective, les champs d'herbe et les champs de feuilles définissent des informations supplémentaires qui sont nécessaires lors de l'évaluation des effets.

Les informations liées aux champs d'herbe sont une *densité* et un *type*. La densité $\mathcal{H}_{dens} \in [0, 1]$ détermine la probabilité d'être instanciées des touffes d'herbes présentes à l'intérieur des champs. Le type m_i est un 4-uplet $\mathcal{H}_{type} = (m_1, m_2, \dots, m_4)$, $\sum_{i=1}^4 m_i = 1, m_i \in [0, 1]$ qui contrôle la probabilité qu'à une touffe d'herbes géométrique instanciée d'être associée à chacun des 4 modèles géométriques prédéfinis (Section 3.4).

De même, les champs de feuille définissent également un *type*, $\mathcal{F}_{type} = (m_1, m_2, \dots, m_{14})$, $\sum_{i=1}^{14} m_i = 1, m_i \in [0, 1]$. Ce 14-uplet fonctionne de la même façon que le type défini par les champs de feuilles et détermine pour une feuille instanciée une probabilité d'être associée à chacun des 14 modèles géométriques prédéfinis (Section 3.5).

3.2 Primitives à squelette

Dans cette partie nous présentons les principaux types de squelettes que nous utilisons pour les différents champs de nos scènes. Les fonctions optimisées de calcul de distance des différents squelettes sont détaillées dans [6] et [122].

3.2.1 Point

Le plus simple de nos champs d'influence est le champ à squelette ponctuel. Il est paramétré par son centre \mathbf{c} et son rayon r . Il est utilisé dans un grand nombre de situations puisqu'il est suffisant pour représenter des caractéristiques très variées : tas de feuilles ou de neige, source de chaleur, etc. (Figure 3.3).

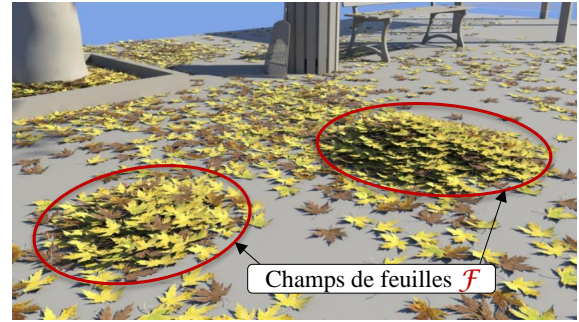
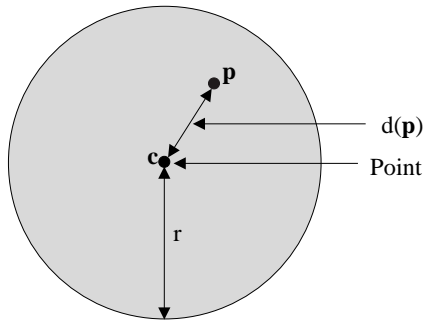


FIGURE 3.3 – Les tas de feuilles de cette scène sont créés à l’aide de champs d’influence à squelette ponctuel.

3.2.2 Segment

Le squelette du second type de champ d’influence est un segment. Ce champ est paramétré par les extrémités **a** et **b** du segment et par un rayon r . Nous utilisons par exemple ce type de champ pour modéliser la chaleur irradiée par un liquide qui circule dans une canalisation, comme illustré dans la figure 3.4.

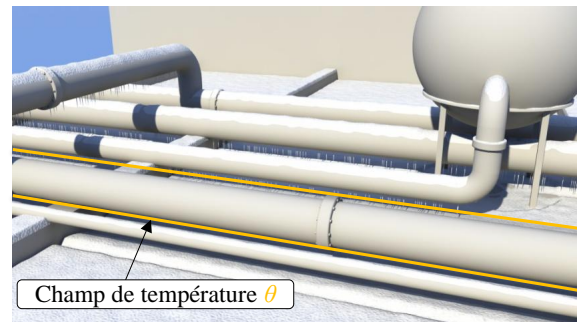
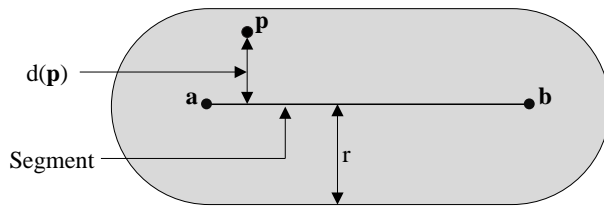


FIGURE 3.4 – La chaleur qui fait fondre la neige et empêche la formation de stalactites autour de ce tuyau est modélisée par un champ de température dont le squelette est un segment.

3.2.3 Courbe

Le squelette de ce champ est défini par une courbe quadrique c construite par morceaux et un rayon r . Ce type de champ permet de représenter des trajectoires telles que celles qui seraient laissées par un véhicule roulant dans la neige, des congères, etc. (Figure 3.5).

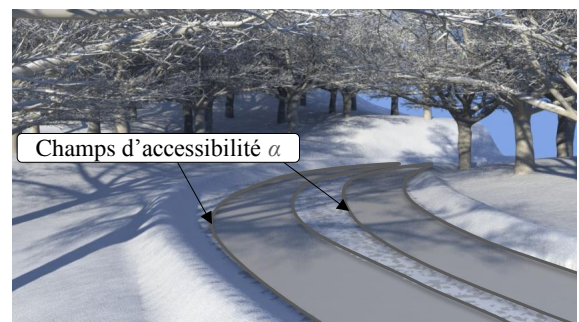
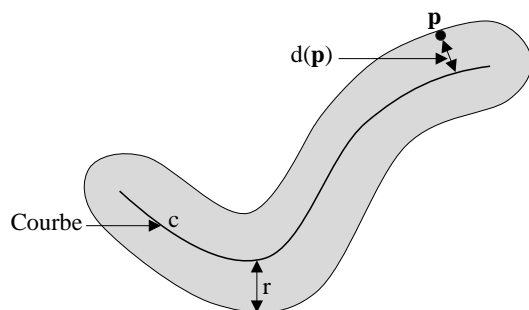


FIGURE 3.5 – Les traces dues au passage des véhicules sont modélisées par des champs d’accessibilité à squelette courbe. Les congères sont créées à l’aide de champs de neige basés sur le même type de squelette.

3.2.4 Cylindre

Ce type de champ d'influence utilise un petit et un grand cylindre dont les extrémités sont \mathbf{a} et \mathbf{b} et les rayons respectivement r_0 et r_1 . Si \mathbf{p} se situe à l'intérieur du petit cylindre d'axe $[\mathbf{a}, \mathbf{b}]$ et de rayon r_0 , la valeur associée à \mathbf{p} est nulle. Si \mathbf{p} est situé à l'intérieur du grand cylindre d'axe $[\mathbf{a}, \mathbf{b}]$ et de rayon r_1 , la valeur est calculée en fonction de la distance au petit cylindre. Ce squelette polyvalent est utilisé pour représenter aussi bien l'accessibilité d'un objet cylindrique (Figure 3.6) que les feuilles générées par un arbre.

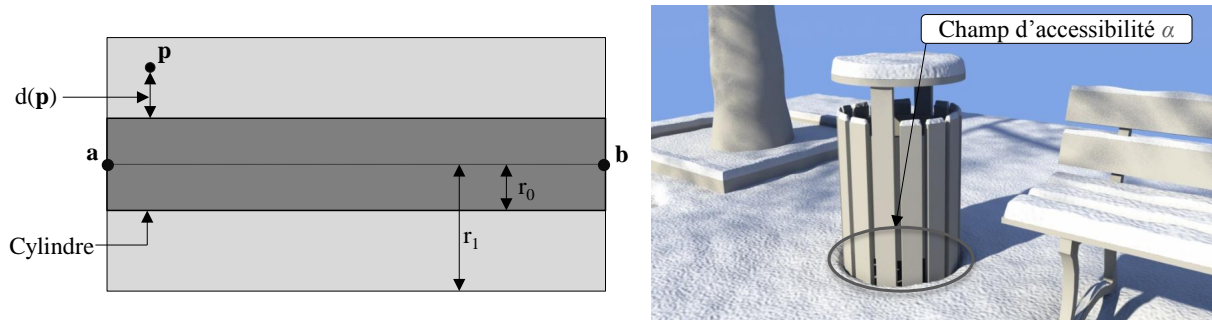


FIGURE 3.6 – La diminution de l'accessibilité sous la poubelle est matérialisée avec un champ à squelette cylindrique.

3.2.5 Boîte

Le principe de ce champ d'influence est proche de celui du champ à squelette cylindrique. Si \mathbf{p} se situe dans la boîte paramétrée par les points extrémaux \mathbf{a} et \mathbf{b} , la valeur associée au phénomène est nulle. Dans le cas contraire, la valeur est calculée en fonction de la distance à la boîte.



FIGURE 3.7 – L'accessibilité qui diminue sous le banc est modélisée par un champ dont le squelette est une boîte.

3.2.6 Comparaison des temps de calcul

Le tableau 3.1 présente les temps nécessaires à un million d'évaluations du calcul de la fonction $d(\mathbf{p})$ en fonction des squelettes des champs d'influence. Les temps de calcul associés aux squelettes point, cylindre, boîte et segment sont similaires et très faibles. La fonction $d(\mathbf{p})$ est beaucoup plus coûteuse dans le cas du squelette courbe, et les temps de calculs sont supérieurs de deux ordres de grandeur.

Squelette	Point	Cylindre	Boite	Segment	Courbe
$d(\mathbf{p})$	3ms	3 ms	4 ms	16ms	869 ms

Tableau 3.1 – Comparaison du temps d'exécution d'un million d'évaluations de $d(\mathbf{p})$ pour les différents squelettes.

3.3 Primitives à base d'image

Contrairement aux champs d'influence précédents, les champs d'influence image n'utilisent pas un objet géométrique comme squelette, mais une image en niveau de gris, projetée sur le plan horizontal de la scène. Introduisons les notations suivantes : \mathbf{p}' est le transformé de \mathbf{p} dans l'espace de l'image, et I est la fonction de l'image renvoyant le niveau de gris d'un pixel. Pour ce type de champ d'influence, le calcul de la distance entre \mathbf{p} et le squelette est remplacé par une évaluation du niveau de gris de l'image au point \mathbf{p}' . La valeur associée à \mathbf{p}' est donnée par le calcul suivant :

$$f(\mathbf{p}) = 1.0 - (1.0 - I(\mathbf{p}')/127)$$

Ce type de champ d'influence affecte donc les phénomènes de la façon suivante :

- $0 \leq I(\mathbf{p}') < 127$: zones sombres de l'image, diminution de la valeur,
- $I(\mathbf{p}') = 127$: zones neutres de l'image, valeur inchangée,
- $128 \leq I(\mathbf{p}') \leq 256$: zones claires de l'image, augmentation de la valeur.

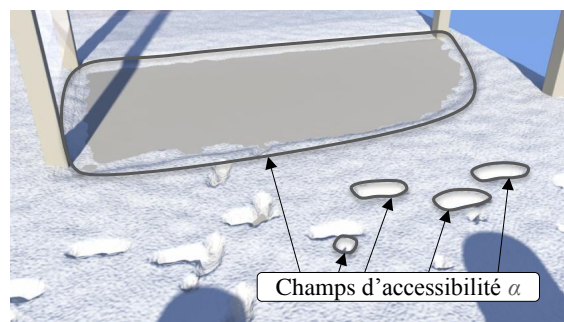
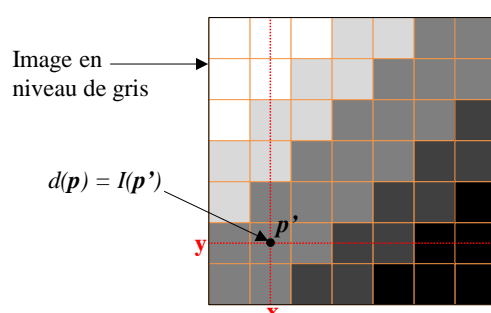


FIGURE 3.8 – La diminution de la hauteur de neige sous l'arrêt de bus et les traces humaines et canines sont modélisées avec des champs à base d'image.

Ce type de champ est utilisé dans des situations très variées, notamment dans les cas où il est nécessaire d'avoir une précision importante. On peut voir une de ces utilisations dans la figure 3.8, où il est utilisé pour modéliser les traces de pas laissées par des piétons dans la neige.

4 Combinaison des champs d'influence

La requête d'évaluation de l'environnement R_e traverse hiérarchiquement l'arbre de construction de la scène afin de quantifier un effet ou un phénomène en un point \mathbf{p} de l'espace. La hiérarchie de volumes englobants associée aux objets environnementaux et aux champs d'influence optimise la requête en évitant les sous parties de l'arbre de construction qui ne contiennent pas de champ associé à l'effet ou au phénomène quantifié.

Il existe une requête d'évaluation de l'environnement pour chaque type de champs. Pour les champs d'effets, les requêtes $\mathcal{N}(p)$, $\mathcal{G}(p)$, $\mathcal{H}(p)$ et $\mathcal{F}(p)$ correspondent respectivement à l'évaluation des champs de neige, de glace, d'herbe et de feuilles. Pour les champs de phénomènes,

les requêtes $\alpha(e)$, $\eta(e)$ et $\theta(e)$ sont respectivement l'évaluation des champs d'accessibilité, d'humidité et de température.

Rappelons que lors de l'évaluation de l'arbre de construction de la scène, la requête de génération des décorations géométriques $R_{\mathcal{E}}$ fait appel à la requête d'évaluation de l'environnement R_e pour quantifier les différents effets et phénomènes (Section 2). Plusieurs champs d'un même type peuvent influencer simultanément \mathbf{p} . Il est donc nécessaire de définir une méthode de combinaison afin que $R_e(\mathbf{p})$ prenne simultanément en compte les différents champs d'un même type qui influencent \mathbf{p} .

4.1 Cas général

L'opération de combinaison des champs d'influence, que ce soient les champs de neige, de glace, d'herbe, de feuilles, d'accessibilité, de température ou d'humidité, doit être commutative, puisque l'on souhaite que l'ordre dans lequel l'arbre de construction est parcouru n'ait pas d'influence sur le résultat de la requête. À l'exception de l'accessibilité, le calcul de la combinaison de plusieurs champs \mathcal{C}_i d'un même type est obtenu en combinant les fonctions f_i de ces champs comme décrit dans le modèle BlobTree [146] :

$$f(\mathbf{p}) = \sum_{i=1}^n f_i(\mathbf{p}). \quad (3.1)$$

La figure 3.9 montre les décorations produites par la combinaison de deux champs d'herbe \mathcal{H}_1 et \mathcal{H}_2 . Ces deux champs ont des paramètres identiques : $\mathcal{H}_1_{dens} = \mathcal{H}_2_{dens} = 0.5$. L'herbe qui s'est développée à l'intérieur d'un des deux champs est disparates et de taille réduite. La situation est différente dans l'intersection $\mathcal{H}_1 \cap \mathcal{H}_2$ des deux champs : la formule de combinaison donne $(\mathcal{H}_1 \cap \mathcal{H}_2)_{dens} = 1.0$. L'herbe qui se développe dans l'intersection des deux champs est deux fois plus dense et plus haute.

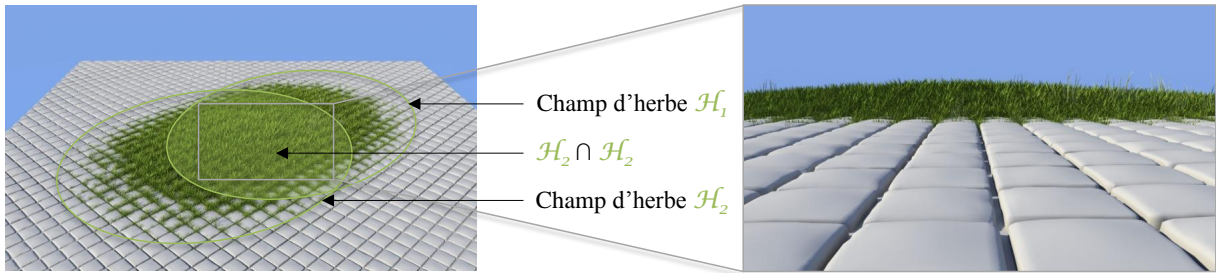


FIGURE 3.9 – Décorations produites par la combinaison de deux champs d'herbe.

4.2 Cas particulier de l'accessibilité

L'accessibilité est un concept issu des travaux de Miller présenté dans [88]. L'accessibilité d'un point de l'espace \mathbf{p} , $\alpha(\mathbf{p})$, mesure la quantité de lumière qu'il peut recevoir. La fonction α renvoie un nombre $a \in [0, 1]$ tel que $a = 1$ si \mathbf{p} est entièrement accessible, $a \in]0, 1[$ si \mathbf{p} est partiellement accessible et $a = 0$ si \mathbf{p} est inaccessible. L'accessibilité de \mathbf{p} est utilisée pour modifier sa composante diffuse lors du calcul de l'éclairage. De cette façon, les zones les moins accessibles d'une scène sont obscurcies lors du rendu, ce qui permet de mieux faire ressortir le relief des objets et permet d'obtenir un rendu plus réaliste. Parmi les techniques de rendu associées à l'accessibilité [81] la plus courante est l'occlusion ambiante (Figure 3.10).

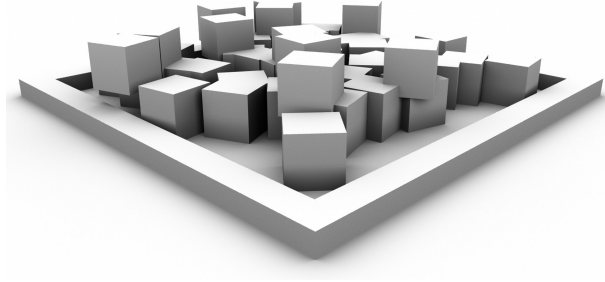


FIGURE 3.10 – Image calculée par occlusion ambiante. Les zones moins accessibles sont obscurcies.

L'accessibilité de \mathbf{p} est définie de la manière suivante :

$$\alpha(\mathbf{p}, n) = \frac{1}{\pi} \int_{\Omega} V(\mathbf{p}, \omega) d\omega \quad (3.2)$$

où α est la fonction d'accessibilité et n la normale à \mathbf{p} . \int_{Ω} représente l'intégration de $V(\mathbf{p}, \omega)$ dans une demie sphère de centre \mathbf{p} orientée en fonction de n . $V(\mathbf{p}, \omega)$ est la fonction de visibilité :

$$V(\mathbf{p}, \omega) = \begin{cases} 1 & \text{si } r(\mathbf{p}, \omega) \cap \mathcal{G} = \emptyset \\ 0 & \text{sinon} \end{cases}$$

$V(\mathbf{p}, \omega)$ retourne 1 si $r(\mathbf{p}, \omega)$, le rayon d'origine \mathbf{p} et de vecteur directeur ω , n'intersecte pas la géométrie G de la scène, et 0 dans le cas contraire.

L'intégrale \int_{Ω} est approximée par le calcul de $V(\mathbf{p}, \omega)$ sur i rayons. Soit $\mathbf{p}_i = \{\mathbf{p}_0, \dots, \mathbf{p}_{i-1}\}$ un ensemble de i points tirés de manière stochastique sur la surface de la demie sphère de centre \mathbf{p} orientée en fonction de n . La fonction d'accessibilité α est souvent implémentée de la manière suivante :

```

1: function  $\alpha(\mathbf{p}, n)$ 
2:    $a := 0$ 
3:   for all  $\omega \in \mathbf{p}_i$  do
4:      $a := a + V(\mathbf{p}, \omega)$ 
5:   end for
6:   return  $a/i$ 
7: end function
```

La figure 3.11 illustre le fonctionnement de l'implémentation classique de l'accessibilité, ramené ici à un exemple en deux dimensions dans un souci de clarté. Huit points sont échantillonnés à la surface de la demi sphère de centre \mathbf{p} orientée en n . Dans l'exemple de gauche \mathbf{p} est entièrement accessible, alors que dans l'exemple de droite il est partiellement accessible. Comme pour toute méthode reposant sur un échantillonnage stochastique, plus le nombre d'échantillons utilisés est important (dans ce cas les i points \mathbf{p}_i tirés à la surface de la demi sphère), plus le calcul sera précis, mais plus il sera lourd. De plus, la fonction de visibilité implique des calculs d'intersection entre des rayons et la géométrie de la scène, et même avec un partitionnement de l'espace adéquat, ces calculs deviennent rapidement coûteux.

Nous avons implémenté cette méthode pour calculer l'impact d'un banc sur une couche de neige. Cependant, tirer des échantillons à la surface d'une demie sphère revient à accorder la même importance à tous les rayons, quelle que soit leur direction. Comme la neige est un phénomène essentiellement vertical, cet échantillonnage n'est pas adapté au calcul de ce phénomène.

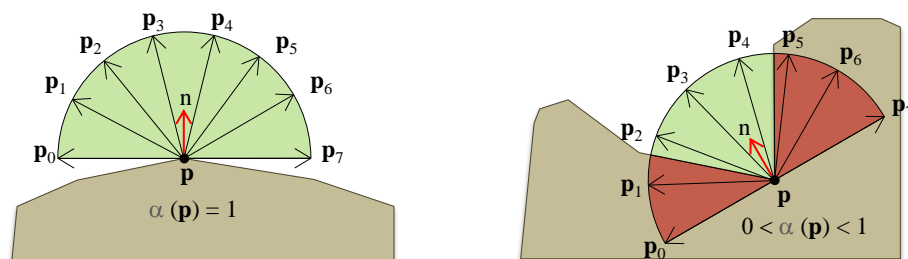


FIGURE 3.11 – Implémentation classique du calcul de l’accessibilité représentée en deux dimensions.

Au lieu de tirer des échantillons à la surface d’une demie sphère de centre \mathbf{p} orientée en n , nous tirons des échantillons à la surface d’un cône de sommet \mathbf{p} orienté en n et d’angle 20 degrés (Figure 3.12). Ce calcul est effectué pour chacun des sommets \mathbf{p} du maillage de neige. Les nouveaux sommets \mathbf{p}' du maillage sont les sommets \mathbf{p} déplacés par la fonction d’accessibilité selon la formule $\mathbf{p}' = (x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}} * \alpha(\mathbf{p}, n))$.

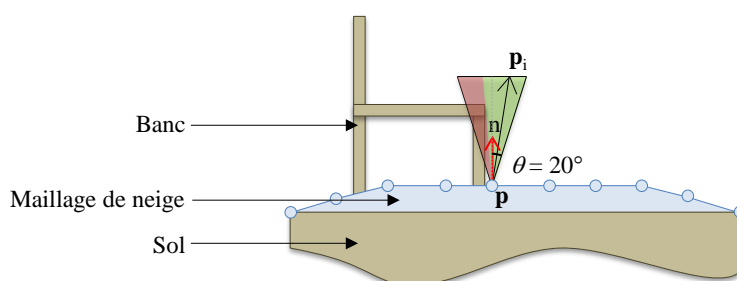


FIGURE 3.12 – Implémentation du calcul d’accessibilité pour la neige représentée en deux dimensions.

Le résultat est visible dans la figure 3.13 : on constate que la quantité de neige est largement diminuée sous le banc. En dessous de 100 échantillons par sommets, le résultat est trop bruité. Avec ce nombre d’échantillons, 1 750 millisecondes sont nécessaires pour calculer l’accessibilité sur ce maillage de neige constitué de 4 511 sommets. Cependant, comme nous voulons un système fonctionnant en temps interactif, nous ne pouvons pas dépendre d’une méthode de calcul aussi coûteuse. Comme pour les autres effets et phénomènes, nous avons choisi de modéliser l’accessibilité par des champs d’influence.



FIGURE 3.13 – Déplacement des sommets du maillage de neige par le calcul d’accessibilité décrit dans la figure 3.12.

Dans la figure 3.14, le calcul est effectué par un champ d’accessibilité. La quantité de neige est plus diminuée sur le banc que dans la figure précédente, et on voit qu’une zone située sous le banc ne comporte plus du tout de neige. Le maillage de neige contient toujours 4 511 sommets,

et cette fois le calcul est effectué en seulement 5 millisecondes : le calcul est donc environ 350 fois plus rapide. Ce résultat prouve l'intérêt de notre approche : même dans le cas de l'accessibilité, un calcul simple et purement géométrique (par opposition aux calculs de phénomènes comme les transferts de chaleur qui impliquent des équations physiques et qui sont beaucoup plus lourds), notre méthode à base de champ d'influence s'avère plus efficace de deux ordres de grandeur.



FIGURE 3.14 – Déplacement des sommets du maillage de neige avec un champ d'accessibilité.

Cependant, la fonction de mélange utilisée pour les autres types de champ d'influence ne convient intuitivement pas au principe de l'accessibilité. L'accessibilité définit un ratio fait pour multiplier une quantité à laquelle elle est associée. Pour que plusieurs champs d'accessibilité puissent fonctionner ensemble, il faut que dans l'intersection de ces champs leurs influences soient simultanément prises en compte. Comme pour la fonction de mélange précédente, la fonction de mélange de l'accessibilité doit être commutative, puisque que les différents champs d'accessibilité ne sont pas ordonnés. Nous la définissons comme suit :

$$\alpha(\mathbf{p}) = \prod_{i=1}^{i=n} \alpha_i(\mathbf{p}). \quad (3.3)$$

La figure 3.15 montre une scène composée de plusieurs objets environnementaux, et en particulier les objets environnementaux *tuyau* et *coude* qui sont dotés d'un champ d'accessibilité. L'image de droite présente un gros plan des tuyaux enchevêtrés, là où leurs champs d'accessibilité s'intersectent et leurs influences se combinent. On distingue clairement que sous plusieurs tuyaux superposés la quantité de neige qui se dépose est fortement diminuée.



FIGURE 3.15 – Combinaison de plusieurs champs d'accessibilité.

5 Accélérations

Dans cette section nous présentons une technique d'optimisation d'évaluation des fonctions de l'environnement : les *opérateurs de niveaux de détails*. Les opérateurs de niveaux de détails sont des nœuds spécifiques de l'arbre de construction qui remplacent les opérateurs de combinaison classique. Leur rôle est d'accélérer les requêtes environnementales en modifiant la traversée de l'arbre. Rappelons que la fonction environnementale $f_{\mathcal{N}}$ d'un nœud \mathcal{N} de l'arbre est réursive :

f_N est la combinaison des fonctions environnementales f_{N_i} de ses fils N_i . Au niveau d'un nœud, l'évaluation de f_N est la combinaison des f_{N_i} et par extension, au niveau de la racine de l'arbre, l'évaluation de f requiert un parcours complet de l'arbre pour prendre en compte la contribution de l'ensemble des champs d'influence de la scène.

Plus N a de fils, plus l'évaluation récursive de f_N devient coûteuse, particulièrement dans le cas où les fonctions f_{N_i} sont complexes à évaluer. Nous avons vu dans la section 3.2 que le coût d'évaluation de la fonction d'un champ dépend du type de son squelette, et qu'il peut être très élevé dans le cas où le squelette est un ensemble de courbes quadratiques. Le coût du calcul de f_N va augmenter avec le nombre de ses fils et la complexité de leurs squelettes.

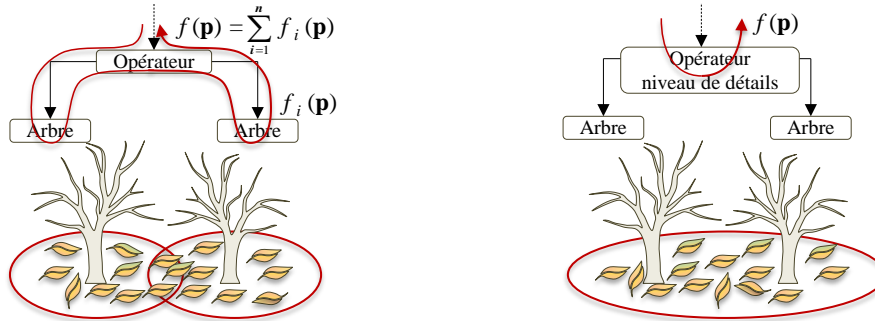


FIGURE 3.16 – Comparaison de l'évaluation de la fonction environnementale entre un nœud classique et un opérateur de niveaux de détails, qui simplifie l'évaluation récursive et qui accélère les requêtes environnementales.

Un opérateur de niveaux de détails N_{nd} simplifie l'évaluation de sa fonction environnementale $f_{N_{nd}}$ en calculant sa valeur localement. Au lieu de traverser récursivement son sous-arbre, ce nœud définit son propre ensemble de fonctions environnementales $f_{N_{nd}}$ (Figure 3.16). Ainsi, pour diminuer le coût de l'évaluation de la fonction environnementale f_N de N , on remplace N par un opérateur de niveaux de détails N_{nd} . Il est ensuite nécessaire de configurer sa fonction $f_{N_{nd}}$ afin qu'elle approche suffisamment f_N . Une fois $f_{N_{nd}}$ configurée correctement, la différence visuelle entre les résultats produits par le nœud original et par l'opérateur de niveaux de détails est négligeable, alors que la vitesse d'évaluation peut être accélérée d'un ordre de grandeur dans certains cas complexes.

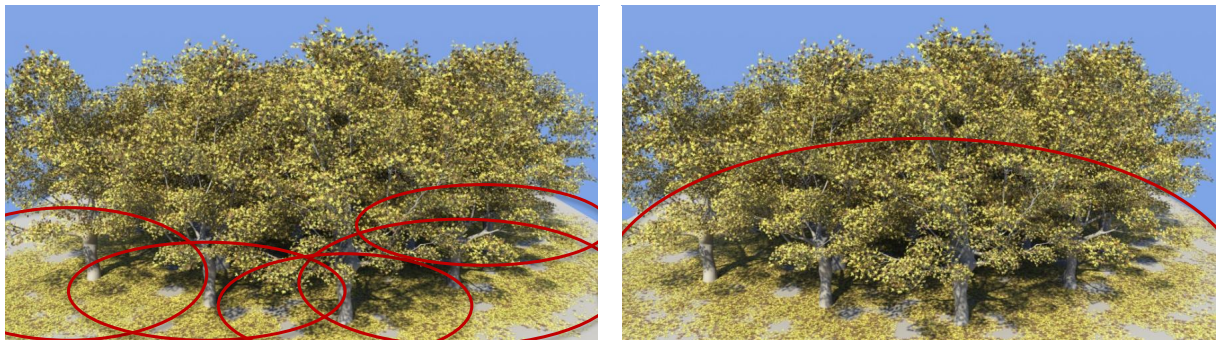


FIGURE 3.17 – Distribution de feuilles obtenue avec une évaluation par objet de la fonction environnementale (gauche) et avec un opérateur de niveau de détail (droite).

La figure 3.17 montre un exemple d'utilisation d'opérateur de niveaux de détails et les statistiques associées sont détaillées dans le tableau 3.2. Cette scène est un bosquet constitué de

vingt platanes, le sol étant une boîte. Dans l'image de gauche, chaque platane génère le champ de feuilles qui lui correspond : l'évaluation de la fonction environnementale se fait par objet, et 200 684 feuilles sont générées au sol en 924 millisecondes. Au contraire, dans l'image de droite, les vingt platanes sont groupés dans un opérateur de niveaux de détails, qui définit un seul champ de feuilles pour l'ensemble de la scène. L'évaluation de la fonction environnementale se fait donc au niveau de l'opérateur de niveaux de détails qui regroupe les platanes, et cette fois 157 741 feuilles sont générées au sol en 378 millisecondes. Il y a une différence négligeable entre ces deux images, mais la deuxième scène est plus facile à contrôler, environ 2,5 fois plus rapide à calculer, et moins coûteuse en ressource puisqu'elle contient seulement 78 % des feuilles de la scène originale. Les opérateurs de niveaux de détails sont donc utiles à la fois pour réduire le temps de calcul d'une scène et pour diminuer sa complexité géométrique.

Opérateur	Nombre de feuilles	Temps de calcul
Combinaison	200 684	924 ms
Niveau de détail	157 741	378 ms

Tableau 3.2 – Comparaison du temps du temps de calcul et du nombre de feuilles générées entre un opérateur de combinaison et un opérateur de niveaux de détails.

6 Conclusion

Nous avons présenté une nouvelle façon de définir l'environnement qui guide un processus de génération dans ce chapitre.

Notre modèle d'environnement repose sur l'utilisation de champs d'influence, des champs scalaires en trois dimensions dont le formalisme est proche de celui des surfaces implicites. Les champs d'influence sont regroupés en deux catégories : les champs d'effets et les champs de phénomènes. La première catégorie est composée de champs qui définissent la valeur d'un effet (neige, glace, herbe, feuilles) alors que la seconde correspond aux phénomènes qui influencent ces effets (accessibilité, chaleur, humidité). Les champs sont combinés par une méthode qui dépend de leur type afin de quantifier spatialement la valeur d'un effet ou d'un phénomène en tout point de la scène.

Ce modèle d'environnement est un outil simple et efficace pour l'utilisateur. Les champs contiennent peu de paramètres, sont simples à configurer, très rapides à évaluer, et leur utilisation est intuitive. Ces sont des outils hautement prédictifs, et quelques champs seulement sont nécessaires pour obtenir une scène complexe, comportant des caractéristiques évoluées.

Cependant, ce modèle pourrait être amélioré sur plusieurs points. De nouveaux types de squelette permettraient de répondre à plus de situation et de rendre l'ensemble plus versatile. Ajouter des champs gérant de nouvelles informations, telles que la vitesse et la direction du vent ou l'ensoleillement, rendrait plus logique et plus naturel le contrôle des effets. L'inconvénient majeur des champs d'influence est qu'ils sont une approximation grossière, et la précision des informations qu'ils fournissent est souvent insuffisante pour s'approcher de la réalité.

Chapitre 4

Effets

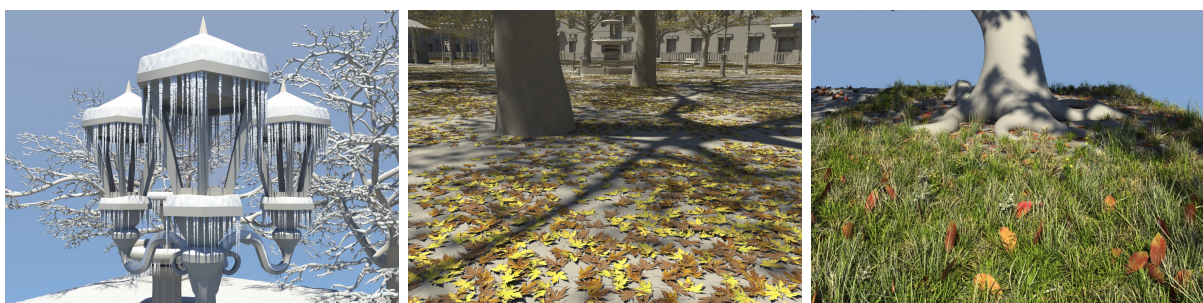


FIGURE 4.1 – Décorations géométriques produites par les effets de notre plateforme. De gauche à droite : neige et glace, feuilles, herbes et feuilles.

Table des matières

1	Introduction	70
2	Neige	70
2.1	Définition des primitives de neige	70
2.2	Fonction d'élévation de neige	71
2.3	Catégories de primitives de neige	73
2.4	Résultats et conclusion	79
3	Glace	81
3.1	Définition des primitives de glace	81
3.2	Sélection, paramétrisation et création du modèle géométrique	82
3.3	Résultats et conclusion	84
4	Herbe	86
4.1	Définition des primitives d'herbe	86
4.2	Les différentes primitives d'herbe	86
4.3	Sélection et paramétrisation des touffes d'herbes candidates	89
4.4	Résultats et conclusion	92
5	Feuilles	94
5.1	Définition des primitives de feuilles	94
5.2	Les différentes primitives de feuilles	94
5.3	Distribution des feuilles candidates	96
5.4	Sélection et paramétrisation des feuilles candidates	98
5.5	Résultats et conclusion	99
6	Accélération	102
7	Conclusion	103

1 Introduction

Dans ce chapitre, nous développons en détail les différents types de décorations géométriques produites par les objets environnementaux et nous proposons de nouvelles techniques pour générer les effets de notre système qui sont la neige, la glace, les feuilles et l’herbe en fonction des paramètres de l’environnement.

Nous présentons une nouvelle approche globale pour dépasser les limitations des méthodes de génération de décorations géométriques. Les méthodes de génération de neige, de glace, de feuilles et d’herbe que nous proposons fonctionnent sur un socle commun : elles utilisent le même format de données, partagent le même formalisme de contrôle, et leurs temps d’exécution sont similaires et interactifs. Elles reposent sur l’utilisation d’effets qui définissent les paramètres nécessaires à la génération des décorations géométriques. Les effets sont des collections de primitives qui s’adaptent à la géométrie des objets environnementaux, définissant les sous-parties de la géométrie qui reçoivent des décorations géométriques et générant ces décorations.

Le choix de notre approche procédurale repose sur le choix de vitesse d’exécution et de contrôlabilité. Nous avons choisi des méthodes efficaces et optimisées afin que leurs temps d’exécution soient faibles et qu’elles soient exploitables dans le cadre de l’édition interactive. Cependant, il est en théorie possible d’utiliser d’autres méthodes plus coûteuses pour produire les décorations géométriques, à partir du moment où l’on peut les contrôler à l’aide de champs d’influence.

2 Neige

Les méthodes de génération de neige que nous avons présentées dans le chapitre 1 partagent toutes les mêmes limitations : elles ne fonctionnent que sur une échelle limitée (elles sont soit capables de produire de la neige sur de grands terrains, soit capables de produire des détails fins, mais sont incapables de gérer simultanément ces deux aspects) et elles sont difficiles à contrôler. Nous proposons une nouvelle approche de génération de neige purement géométrique intégrée à notre système fonctionnant à partir de primitives de neige. Cette méthode est rapide et permet de gérer des détails de l’ordre du centimètre sur des scènes de plusieurs milliers de mètres carrés en seulement quelques secondes. Le contrôle de la génération de la neige se fait à l’aide de champ de neige qui définissent le volume des précipitations, ainsi qu’à l’aide des champs d’accessibilité qui diminuent la hauteur de neige de façon relative et de champs de température qui diminuent la quantité de neige de façon absolue.

2.1 Définition des primitives de neige

Une *primitive de neige*, notée \mathcal{P}_N , est un maillage triangulaire paramétré texturé dont la géométrie et la discrétisation s’adaptent à l’échelle et à la géométrie \mathcal{G} de l’objet environnemental auquel elle est associée (Figure 4.2). Chaque sommet \mathbf{p}_i du maillage est associé à un vecteur de déplacement \mathbf{v}_i . Les sommets \mathbf{p}_i sont déplacés depuis leur position d’origine \mathbf{p}_o dans la direction de \mathbf{v}_i par une fonction d’élévation $\delta(\mathbf{p}_o)$ qui est dépendante de $d(\mathbf{p}_o)$, la fonction de distance entre le bord de \mathcal{P}_N et \mathbf{p}_o .



FIGURE 4.2 – Les sommets de la primitive de neige sont déplacés pour générer la surface de neige.

Chaque primitive de neige dispose d'un paramètre h_{max} qui permet de définir, quand cela est nécessaire, une quantité maximale de neige qu'elle peut recevoir. Dans l'exemple de gauche de la figure 4.3, la valeur de h_{max} est infinie, ce qui conduit à une anomalie visuelle puisque cette primitive reçoit une hauteur de neige bien trop importante par rapport à sa surface. Dans l'exemple de droite, h_{max} vaut 1 centimètre, le maillage de cette primitive est alors plus cohérent.

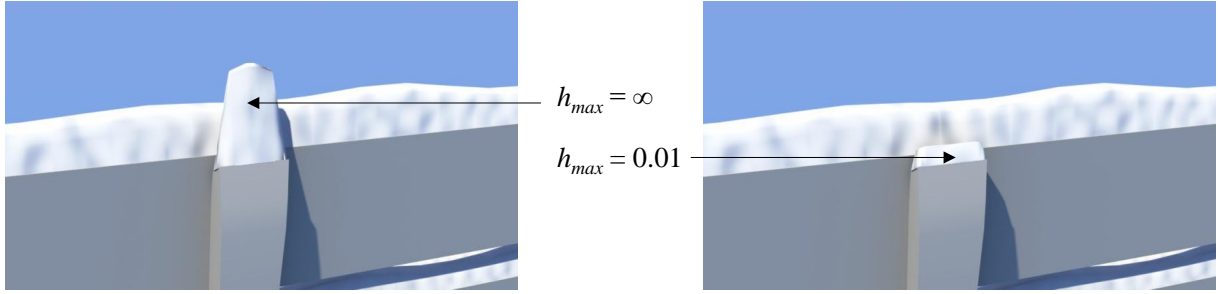


FIGURE 4.3 – Le paramètre h_{max} est utilisé pour limiter la hauteur de neige générée par des primitives dont la surface est très faible.

2.2 Fonction d'élévation de neige

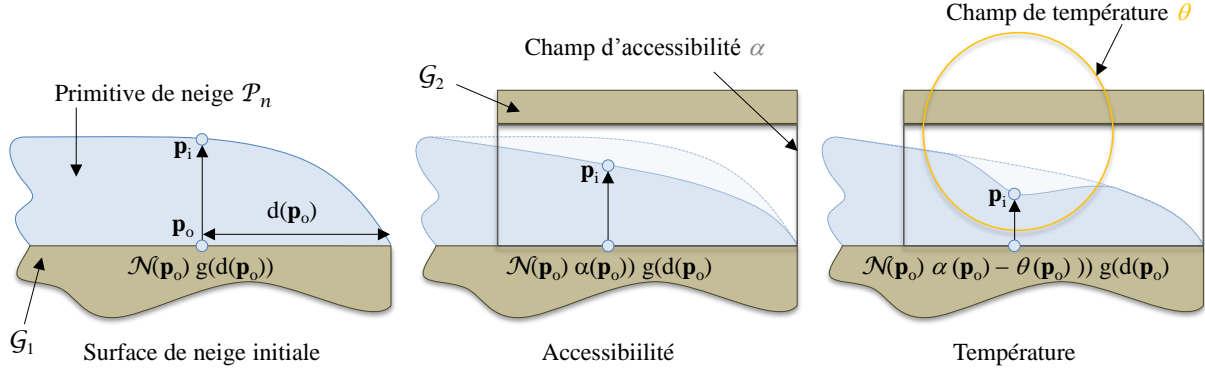
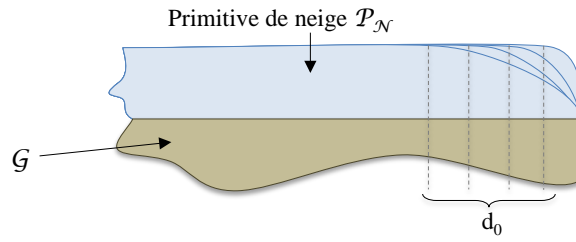
La fonction d'élévation de neige δ est illustrée dans la figure 4.4. Elle dépend de plusieurs paramètres de l'environnement : la hauteur de neige $\mathcal{N}(\mathbf{p}_o)$, l'accessibilité $\alpha(\mathbf{p}_o)$, et la température $\theta(\mathbf{p}_o)$. Elle dépend aussi de $d(\mathbf{p}_o)$, la distance de \mathbf{p}_o au bord de \mathcal{P}_N . Nous utilisons également une fonction de bruit b pour perturber légèrement le déplacement \mathbf{p}_o , afin d'obtenir une couche de neige plus naturelle et moins lisse. Formellement, la fonction d'élévation $\delta(\mathbf{p}_o)$ s'écrit de la manière suivante :

$$\delta(\mathbf{p}_o) = \min(h_{max}, (\mathcal{N}(\mathbf{p}_o)\alpha(\mathbf{p}_o) - \theta(\mathbf{p}_o) - b(\mathbf{p}_o))g(d(\mathbf{p}_o)))$$

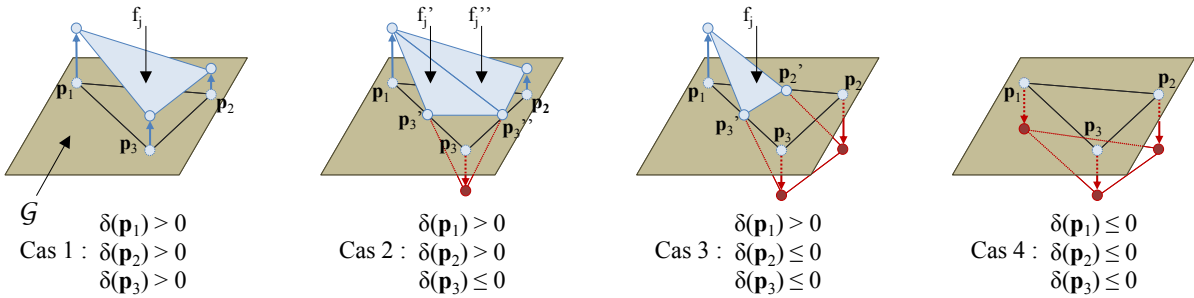
La fonction de neige $\mathcal{N}(\mathbf{p}_o)$ renvoie une valeur positive ou nulle qui quantifie les précipitations de neige en \mathbf{p}_o . La fonction d'accessibilité $\alpha(\mathbf{p}_o)$ retourne une valeur $a \in [0, 1]$ qui diminue la hauteur de neige de façon relative par rapport à l'accessibilité de s_{i_0} . La fonction de température $\theta(\mathbf{p}_o)$ renvoie une valeur $t \in \mathbb{R}$ qui quantifie la chaleur en \mathbf{p}_o et diminue la hauteur de neige de manière absolue. La fonction de distance $d(\mathbf{p}_o)$ calcule la distance entre le bord de la primitive et \mathbf{p}_o . Pour gérer le profil courbe de la surface de neige au bord des primitives, nous utilisons la fonction g définie ci dessous :

$$g(d) = \begin{cases} (1 - (1 - d/d_0)^2)^4 & \text{si } d < d_0 \\ 1 & \text{sinon} \end{cases}$$

L'influence du paramètre d_0 sur le profil de la surface de \mathcal{P}_N au bord de l'objet est illustré dans figure 4.5.


 FIGURE 4.4 – Calcul de la fonction d'élévation de neige δ .

 FIGURE 4.5 – Influence du paramètre d_0 sur le profil de \mathcal{P}_N .

Comme $\theta(\mathbf{p}_o)$ diminue la hauteur de neige de manière absolue, il est possible que dans les zones où la température est élevée, le résultat de $\delta(\mathbf{p}_o)$ soit négatif, ce qui implique que \mathbf{p}_i sera déplacé en dessous de \mathbf{p}_o . Pour pallier ce problème, un traitement postérieur à l'élévation de neige est effectué sur le maillage de \mathcal{P}_N (Figure 4.6). Soit $f_j(s_1, s_2, s_3)$ une face de \mathcal{P}_N dont les trois sommets sont s_1, s_2 et s_3 . En éliminant les cas symétriques, il existe quatre cas (Figure 4.6).


 FIGURE 4.6 – Post-traitement de l'élévation de neige afin d'empêcher le déplacement des sommets \mathbf{p}_i sous leur position d'origine \mathbf{p}_o .

Le cas 1 est trivial : tous les sommets de f_j ont subi un déplacement positif, la face est donc conservée telle quelle. Dans le cas 2, seul le sommet \mathbf{p}_3 a subi un déplacement négatif. Deux nouveaux sommets situés sur la surface de \mathcal{G} , sont calculés par interpolation linéaire inverse : \mathbf{p}'_3 entre \mathbf{p}_1 et \mathbf{p}_3 et \mathbf{p}''_3 entre \mathbf{p}_2 et \mathbf{p}_3 . f_j est supprimée et deux nouvelles faces $f'_j(\mathbf{p}_1, \mathbf{p}'_3, \mathbf{p}_2)$ et $f''_j(\mathbf{p}_1, \mathbf{p}''_3, \mathbf{p}_2)$ sont ajoutées à \mathcal{P}_N . Dans le cas 3 les sommets \mathbf{p}_2 et \mathbf{p}_3 ont été déplacés sous \mathcal{G} . De la même façon que dans le cas précédent, deux nouveaux points \mathbf{p}'_2 et \mathbf{p}'_3 sont calculés. La face f_j est ensuite mise à jour et est maintenant $f_j(\mathbf{p}_1, \mathbf{p}'_2, \mathbf{p}'_3)$. Le cas 4 est le second cas trivial : tous les sommets ont subi un déplacement négatif ou nul, f_j sera donc supprimée.

2.3 Catégories de primitives de neige

Nous définissons deux catégories de primitives de neige : les primitives de neige géométriques et les primitives de neige procédurales (Figure 4.7). Les primitives de neige géométriques sont extraites du maillage de l'objet et liées à ce dernier, et il n'existe qu'une primitive géométrique par objet. Au contraire, les primitives de neige procédurales sont faites pour être utilisées avec des formes géométriques simples et ne sont pas liées à une géométrie spécifique. Dans de nombreux cas, plusieurs primitives procédurales sont utilisées pour enneiger un même objet.

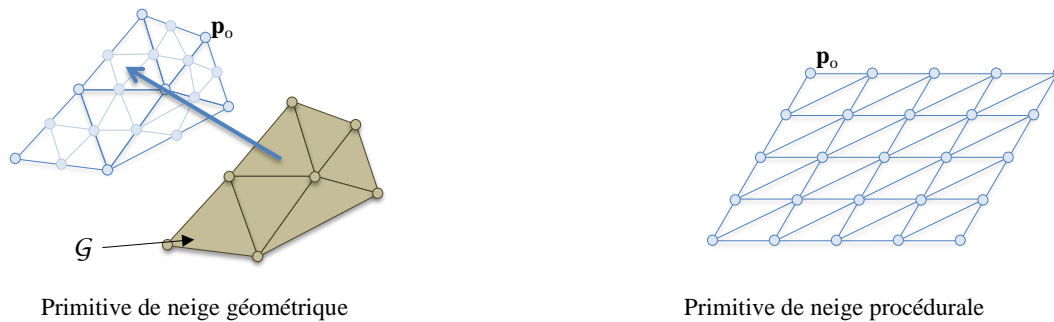


FIGURE 4.7 – Les primitives géométriques sont extraites de la géométrie \mathcal{G} , tandis que les primitives procédurales correspondent à des formes géométriques simples.

2.3.1 Primitives de neige géométriques

Les primitives de neige géométriques sont extraites de la géométrie \mathcal{G} de l'objet environnemental auquel elles sont associées. Elles sont utilisées pour modéliser la neige dans les cas où \mathcal{G} est complexe ou très fine, telle que des branches d'arbre ou du grillage (Figure 4.8). Elles peuvent être progressivement raffinées pour adapter le maillage de neige à la précision souhaitée, ce qui permet de traiter efficacement de grandes scènes ou de gérer le niveau de détail des maillages de neige associés.

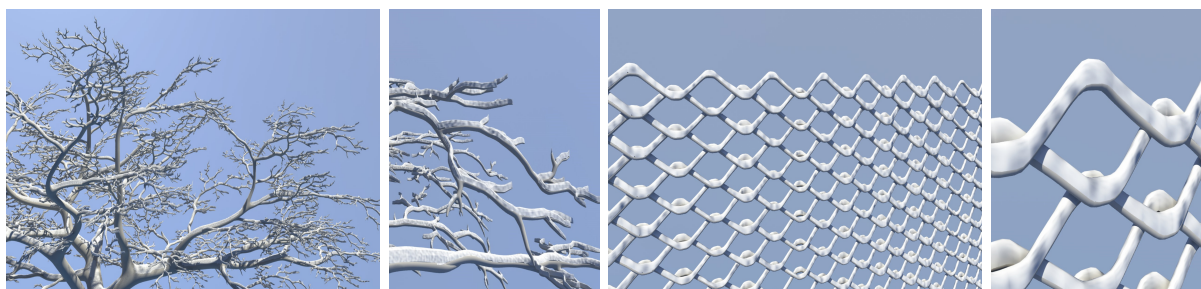


FIGURE 4.8 – Les primitives de neige géométriques sont adaptées aux objets complexes et aux détails fins (petites branches, grillages, fils électriques, etc.).

Soit f_i les faces de \mathcal{G} , \mathbf{n}_i les vecteurs normaux aux faces, s_{max} un seuil de pente défini par l'utilisateur, \mathbf{z} le vecteur vertical unitaire, et δ' une fonction d'élévation. Les primitives géométriques sont générées à partir de \mathcal{G} en trois étapes : création d'une collection de faces $C_f = \cup_{f_i} \mid \mathbf{n}_i \cdot \mathbf{z} > s_{max}$, subdivision de C_f , puis élévation des sommets de C_f .

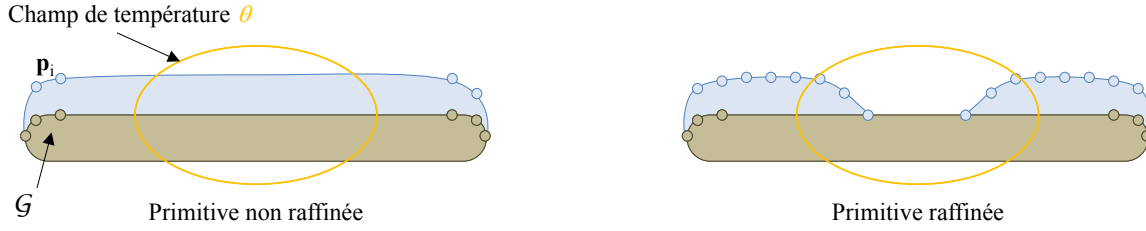


FIGURE 4.9 – Une géométrie \mathcal{G} trop grossière ne permet pas aux primitives de neige géométriques de capturer correctement les informations de l’environnement.

La première étape consiste à créer une collection de faces C_f en parcourant \mathcal{G} et en retenant uniquement les f_i dans le cas où $\mathbf{n} \cdot \mathbf{z} > s_{max}$. De cette façon, on s’assure de ne conserver que les faces de \mathcal{G} qui sont tournées vers le haut, et dont la pente est inférieure à un seuil donné. Une fois cette étape terminée, si la géométrie des faces est trop grossière, en particulier aux bords, nous utilisons une méthode de subdivision pour raffiner le maillage lors de la seconde étape (Figure 4.9). Les primitives de neige peuvent être raffinées progressivement jusqu’à obtenir la précision désirée. Cette fonctionnalité est utile dans le cas où l’influence d’un champ n’est pas capturée à cause d’une géométrie trop grossière (Figure 4.10).

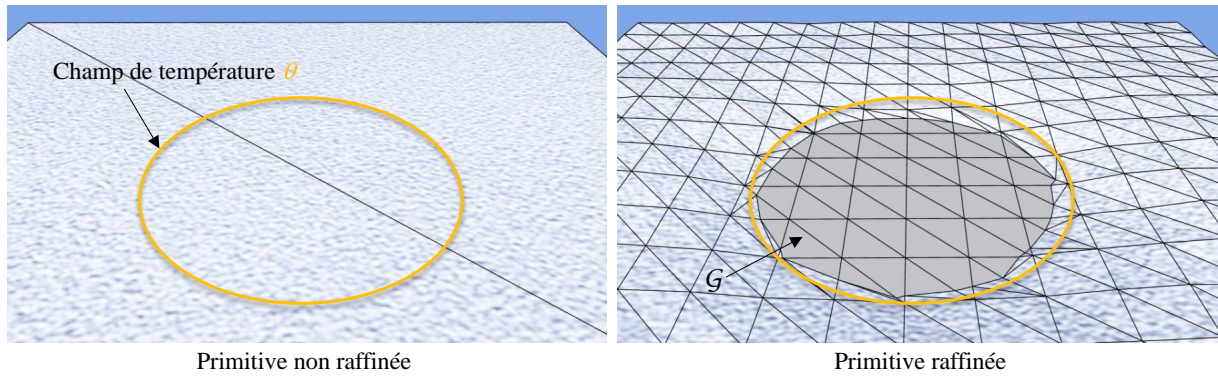


FIGURE 4.10 – Dans l’exemple de gauche, \mathcal{G} est trop grossière, la primitive de neige géométrique associée ne pourra donc pas capturer la chaleur produite par le champ de température θ . Dans l’exemple de droite, la primitive de neige géométrique suffisamment raffinée peut prendre en compte les champs d’influence correctement.

La dernière étape consiste à élever les sommets de C_f , en s’appuyant sur la fonction d’élévation de neige δ (Section 2.2). Toutefois, dans le cas des primitives géométriques, la fonction d’élévation est adaptée pour à la fois prendre en compte l’inclinaison des faces (et ainsi distribuer plus de neige sur les faces les plus horizontales) et garantir leur raccord. La fonction d’élévation utilisée pour les sommets des primitives géométriques est définie par :

$$\delta'(\mathbf{p}_i) = \delta(\mathbf{p}_o) \mathbf{n}_i \cdot \mathbf{z}$$

De cette façon, les sommets du maillage de la primitive géométrique s’adaptent parfaitement à la géométrie de l’objet \mathcal{G} étant donné qu’ils sont issus de cette géométrie. Nous nous assurons donc de ne jamais voir apparaître de décalages ou de fissures entre le maillage de neige et \mathcal{G} .

2.3.2 Primitives de neige procédurales

Contrairement aux primitives de neige géométriques, les primitives de neige procédurales sont des primitives paramétrées spécialisées et optimisées. Elles sont utilisées pour générer une

couche de neige sur des formes géométriques simples comme des boîtes, des cylindres, des sphères ou des surfaces de révolution qui sont courantes dans l'architecture moderne et le mobilier urbain (Figure 4.11). Elles génèrent le maillage de neige depuis un ensemble de paramètres définissant leur forme et leur niveau de subdivision. Elles sont également utilisées dans le cadre des objets environnementaux paramétrés, où les primitives des différents sous-objets doivent être configurées avec le même niveau de précision et partager les mêmes conditions aux bords pour se raccorder correctement (Annexe A).

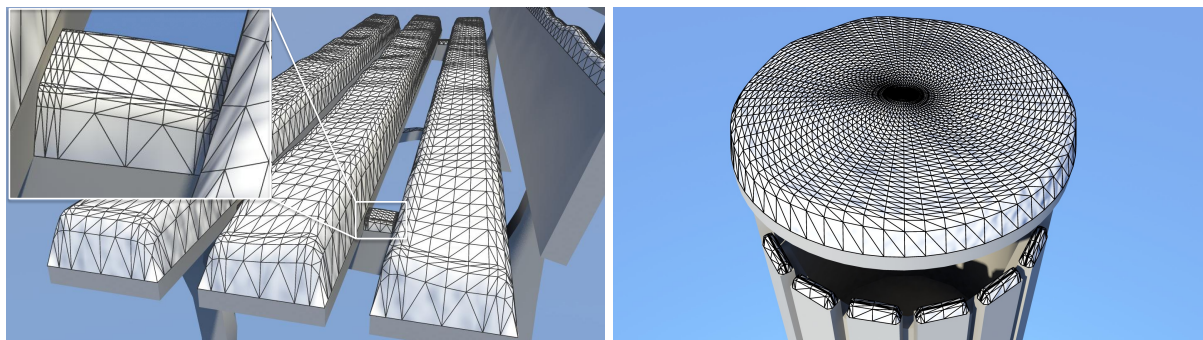


FIGURE 4.11 – Les primitives de neige procédurales sont adaptées aux formes géométriques simples.

Différents types de primitives géométriques ont été nécessaires pour produire les différentes scènes : nous avons créé des primitives procédurales quadrangulaires, discoïdes, sphériques, de révolution, et basées sur une grille. Ces différentes primitives s'adaptent chacune à un type de géométrie spécifique.

Primitive procédurale quadrangulaire Les primitives procédurales quadrangulaires sont conçues pour créer un maillage de neige sur un polygone quadrangulaire convexe dont les sommets ne sont pas nécessairement coplanaires (Figure 4.12). Les paramètres nécessaires à la création d'une primitive procédurale quadrangulaire sont les sommets du polygone $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, les pentes aux sommets s_0, s_1, s_2, s_3 , les pentes aux arêtes a_0, a_1, a_2, a_3 , et la discrétisation d .

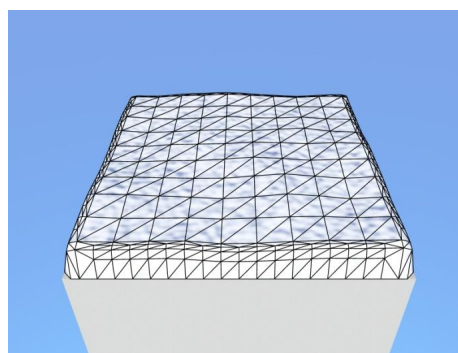
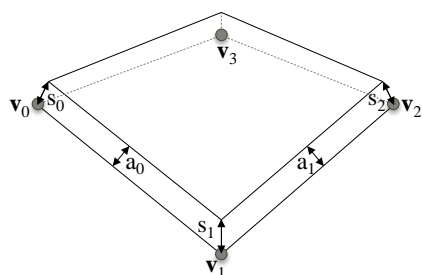


FIGURE 4.12 – Maillage produit par une primitive procédurale quadrangulaire.

Cette primitive très polyvalente est utilisée dans un grand nombre de situations pour enneiger des objets environnementaux très variés tel que du mobilier, des portes, des fenêtres, des toits ou des bâtiments (Figure 4.13). Comme le comportement du maillage aux bords est variable grâce aux paramètres de pente aux sommets et aux arêtes, elle est particulièrement bien adaptée aux objets environnementaux paramétrés.



FIGURE 4.13 – Les escaliers, les bancs, la table, les fenêtres, les portes, les piliers et le toit sont enneigés avec des primitives procédurales quadrangulaires.

Primitive procédurale discoïde Les primitives procédurales discoïdes sont utilisées pour créer une couche de neige sur des disques (Figure 4.14). La création d'une primitive discoïde requiert les paramètres suivants : le centre \mathbf{c} , le rayon r , et la discrétisation d .

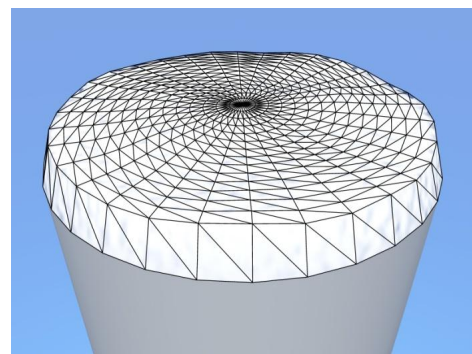
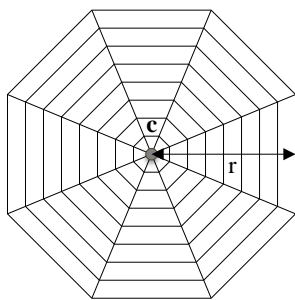


FIGURE 4.14 – Les primitives procédurales discoïdes sont utilisées pour générer de la neige à la surface de disques.

Cette primitive est principalement utilisée pour du mobilier urbain, sa forme la rendant peu polyvalente (Figure 4.15).

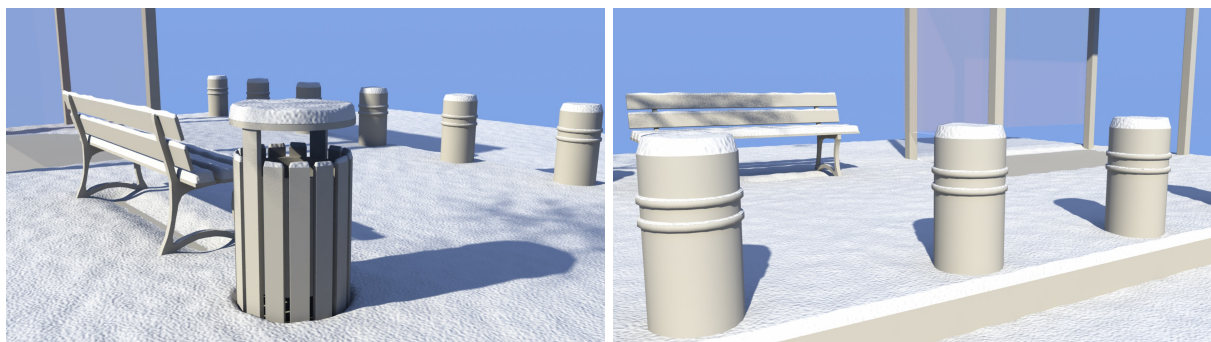


FIGURE 4.15 – La poubelle et les bornes sont enneigées avec des primitives procédurales discoïdes.

Primitive procédurale sphérique Les primitives procédurales sphérique sont utilisées pour créer la surface de neige d'une sphère (Figure 4.16). La création d'une primitive sphérique nécessite les paramètres suivants : le centre \mathbf{c} , le rayon r , et la discrétisation d .

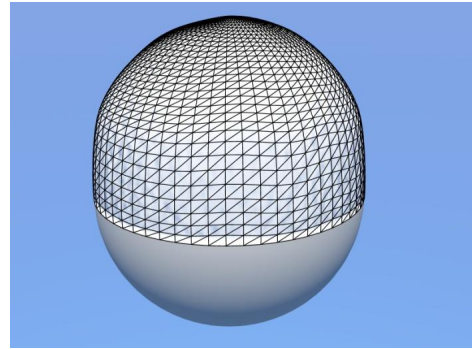
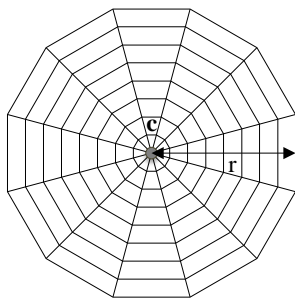


FIGURE 4.16 – Les primitives procédurales sphériques sont conçues pour modéliser de la neige sur des sphères.

Cette primitive est également majoritairement utilisée pour du mobilier urbain, puisque sa forme très spécifique ne la rend pas polyvalente (Figure 4.17).

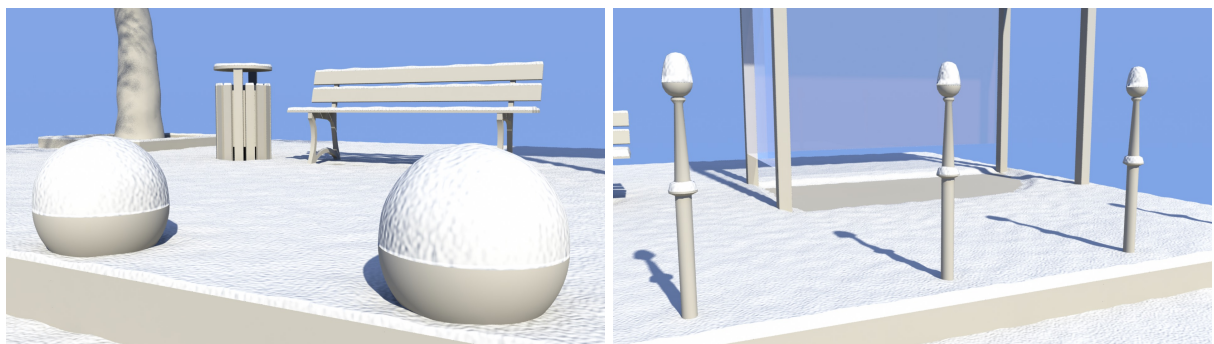


FIGURE 4.17 – Les bornes et les poteaux sont recouverts de neige à l'aide de primitives procédurales sphériques.

Primitive procédurale de révolution Les primitives procédurales de révolution sont utilisées pour créer une couche de neige sur une surface de révolution. Les paramètres nécessaires à leur construction sont le centre c , l'ensemble des points p_i définissant la surface de révolution, la présence ou l'absence d'un trou entre le dernier point du profil et le centre t , et la discrétisation d . Ce type de primitive est adapté à des situations très variées. Nous l'utilisons pour générer la couche de neige qui se dépose sur des surfaces de révolutions, comme les lampadaires, les poteaux, etc. (Figure 4.18).

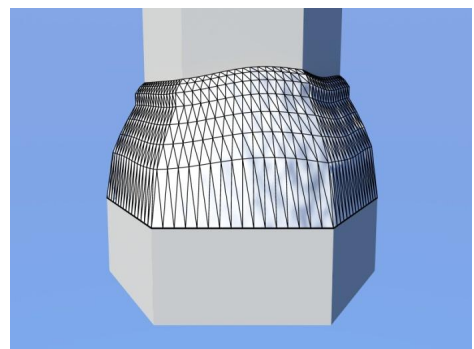
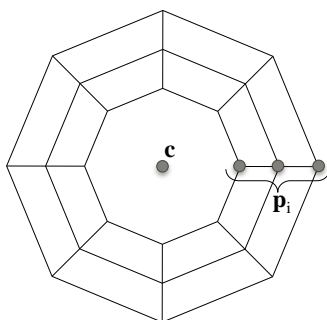


FIGURE 4.18 – Les primitives procédurales de révolution sont conçues pour enneiger des surfaces de révolution.

Primitive procédurale à grille Les primitives procédurales à grille créent une surface de neige sur une grille en trois dimensions. Les paramètres nécessaires à leur création sont le nombre de lignes l , le nombre de colonnes c , et l'ensemble des points de la grille \mathbf{p}_i . Cette primitive est utilisée quand la géométrie à enneiger n'est pas proche d'une forme géométrique de base et que l'utilisation d'une primitive géométrique est exclue. Ce cas se produit souvent pour des éléments fins de géométrie de petite taille, tels que les hampes des lampadaires ou les barrières (Figure 4.19).

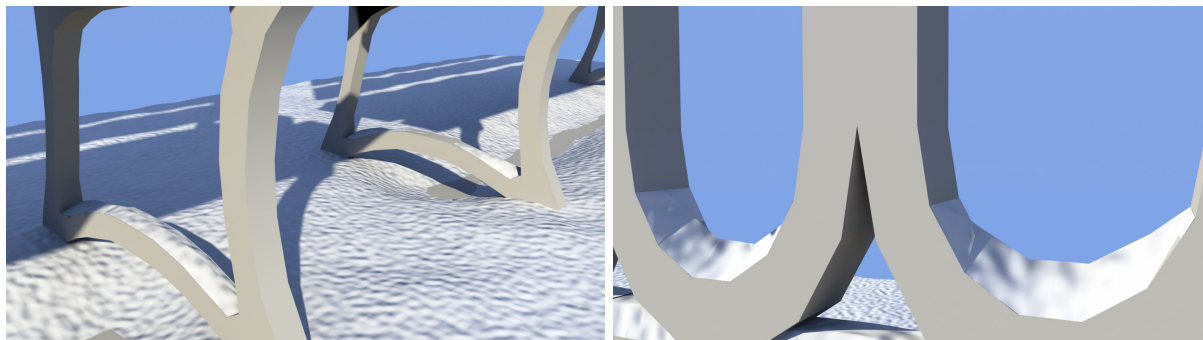


FIGURE 4.19 – Le pied du banc et l'intérieur des barrières utilisent des primitives procédurales à grille pour créer leur couche de neige.

2.3.3 Comparaison entre primitives géométriques et procédurales

Comme les primitives de neige géométriques sont extraites directement depuis la géométrie \mathcal{G} de l'objet environnemental auquel elles sont associées et qu'elles couvrent un grand nombre de cas d'utilisation, la question de la légitimité de l'utilisation des primitives procédurales peut se poser. Cependant, les primitives géométriques ne répondent pas à tous les cas d'utilisation : la géométrie de l'objet auquel elles sont associées doit être pensée pour être utilisée avec ce type de primitive. Dans le cas contraire, le résultat sera inexploitable à cause du grand nombre d'artefacts.

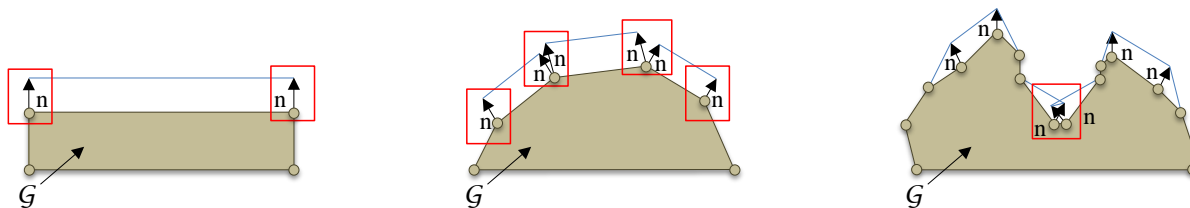


FIGURE 4.20 – Exemple de géométries non adaptées à l'utilisation de primitives géométriques. Les zones encadrées en rouge sont celles où se produisent des artefacts.

La figure 4.20 illustre différents exemples de géométrie qui ne conviennent pas à l'utilisation de primitives de neige géométriques. Le premier cas montre que lorsque \mathcal{G} est trop simple, la surface de neige générée ne se raccorde pas aux bords de la primitive. Dans le second cas, les normales des sommets de \mathcal{G} sont identiques aux normales des faces auxquelles ils appartiennent. Les faces du maillage de neige générée ne se raccordent pas entre elles et ne sont pas non plus connectées à \mathcal{G} . Le dernier cas montre des auto-intersections qui apparaissent sur le maillage de neige lorsque \mathcal{G} est concave.

2.4 Résultats et conclusion

Nous présentons dans cette partie des scènes enneigées produites par notre système. Les statistiques de ces résultats sont données dans le tableau 4.1. Les temps d'exécution fournis dans ce chapitre ont été mesurés sur une machine équipée d'un processeur *Intel Core i7* cadencé à 3.4 GHz et pourvue de 16 Go de mémoire.

Scène	Temps (s)	Triangles (M)	Taille (m)	Détails (cm)
Montmartre (Figure 4.21)	6.02	9.39	33 x 100	1 - 10
Parc (Figure 4.22)	2.80	18.06	115 x 90	1 - 20
Raffinerie (Figure 4.23(a))	0.11	0.32	10 x 11	1 - 6
Square (Figure 4.23(b))	2.70	10.16	10 x 10	1 - 10

Tableau 4.1 – Statistiques de la génération de neige. Le temps de génération est exprimé en secondes, le nombre de triangle de neige en millions, la taille des scènes en mètres et les détails en centimètres.

Nous avons modélisé un quartier d'environ 3 300 mètres carrés semblable à Montmartre (Figure 4.21). La neige de ce quartier, qui représente environ 9.39 millions de triangles, est générée en 6.02 secondes. Ce temps de génération, qui peut paraître élevé, est dû à la très grande quantité de volumes d'accessibilité qui sont utilisés dans cette scène pour modéliser aussi bien des traces de pas (visibles entre autre au niveau des escaliers) que des traces de véhicules. Sans ces volumes qui génèrent des petits détails, la neige de ce quartier est générée en 1.40 secondes seulement.



FIGURE 4.21 – Un quartier semblable à Montmartre sous la neige, comportant de nombreuses traces dues au passage de piétons et de véhicules.

Avec une superficie de 10 350 mètres carrés, le parc (Figure 4.22) est la plus grande des scènes que nous avons créées. Sa couche de neige est composée de plus de 18 millions de triangles. Cette quantité élevée est notamment due au fait que le parc comporte 18 platanes et que chaque platane génère à lui seul environ 344 000 polygones de neige.

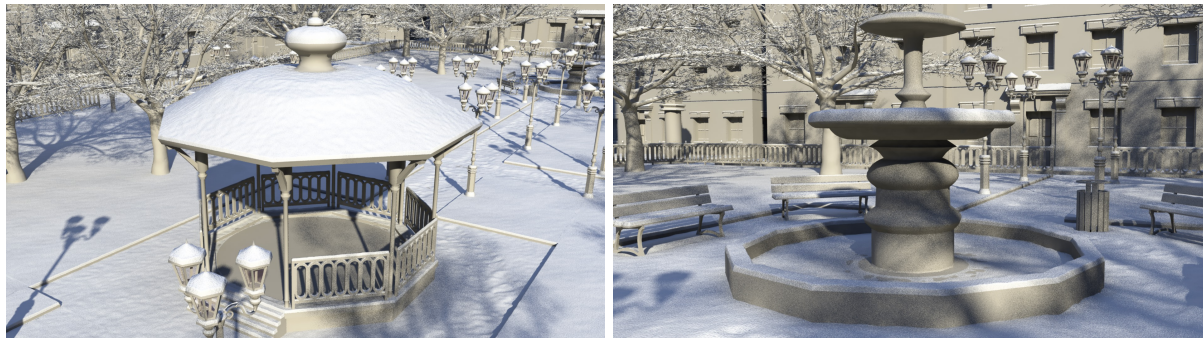


FIGURE 4.22 – Le parc en hiver. Les allées et les zones à proximité du kiosque et de la fontaine ont été piétinées par des passants.

La raffinerie (Figure 4.23(a)) est une scène beaucoup plus petite, qui mesure 110 mètres carrés. Les objets environnementaux qui la composent sont géométriquement simples, comportent peu de primitives géométriques, et sont peu nombreux. En conséquence, la totalité du maillage de neige de cette scène n'est composée que de 320 000 triangles, et il est généré en seulement 0.11 secondes. Le square (Figure 4.23(b)) est également une petite scène, puisqu'elle mesure 100 mètres carrés. Contrairement à la scène précédente, il comporte un grand nombre d'objets environnementaux (chaque pavé de la route est par exemple un objet environnemental) dont certains ont une géométrie très complexe (arbre, statue) et utilisent des primitives de neige géométriques. Le maillage de neige est composé de 10.16 millions de triangles (ce qui est un nombre élevé pour une scène de cette taille) et requiert 2.70 secondes de calcul.

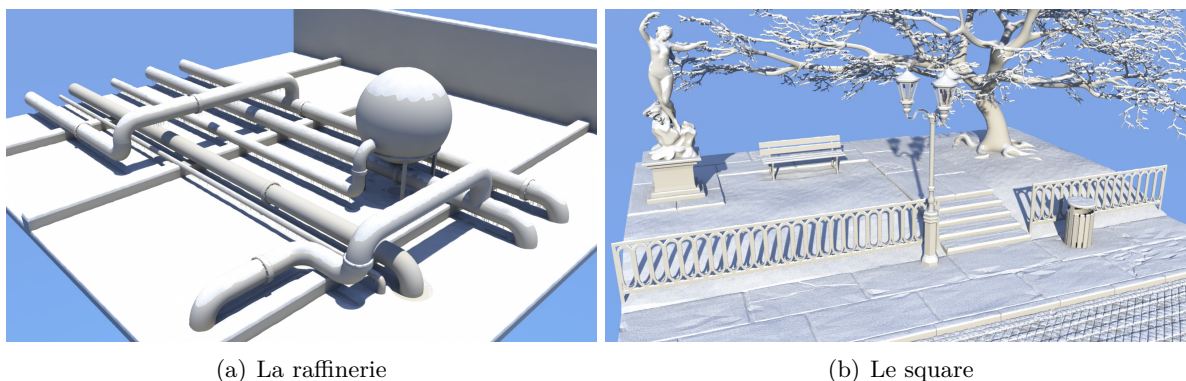


FIGURE 4.23 – Deux scènes de petite taille. La raffinerie est une scène simple qui contient peu d'objets et de détails. Le square, au contraire, contient plusieurs centaines d'objets complexes et beaucoup de détails fins.

Conclusion

Les méthodes de simulation qui utilisent une division de l'espace en grille régulière sont limitées par plusieurs aspects qui sont liés : la qualité des résultats, le coût en mémoire et les temps de calcul. Pour produire de bons résultats, il est nécessaire d'utiliser une résolution très importante. Cependant, comme l'espace mémoire et les temps d'exécution augmentent avec la résolution, il est impossible de générer de grandes scènes avec cette famille de méthodes. Les méthodes de simulation par particules sont très coûteuses et peuvent nécessiter plusieurs heures de calcul pour produire un résultat. Les méthodes de génération par déplacement de surfaces permettent d'enneiger de grandes étendues de terrains, mais ne sont pas adaptées pour générer

des détails inférieurs au mètre.

La méthode de génération de neige que nous proposons résout en partie ces problèmes. Comme les primitives de neige sont liées aux objets environnementaux, il n'existe pas de contrainte de position ou d'alignement des objets dans la scène. Nous sommes capables de générer de grandes scènes de plusieurs milliers de mètres carrés comportant des détails inférieurs au centimètre en quelques secondes seulement.

Cette méthode reste cependant sujette à plusieurs limitations. La géométrie des objets environnementaux doit être créée en prenant en compte les contraintes liées à l'utilisation des différentes primitives de neige. Les résultats produits par la méthode sont crédibles quand ils sont observés à quelques mètres de distance, mais les imperfections des maillages de neige générés par cette méthode deviennent visibles lorsque les résultats sont observés de près. Cette méthode fonctionne bien avec des hauteurs de neige faible, de quelques centimètres. Cependant, les résultats sont de moins bonne qualité avec des hauteurs de neige importantes, vu qu'elle ne gère pas la fusion entre les maillages de neige des différents objets environnementaux.

3 Glace

Nous avons passé en revue plusieurs méthodes de génération de glace dans l'état de l'art (Chapitre 1). Toutes ces méthodes sont conçues pour gérer la croissance de glace sur de petits objets (verres, vitraux, feuilles, branches, etc.) avec une précision de l'ordre du millimètre. La seule méthode qui fait exception est celle de [77] qui fonctionne sur de petits terrains et dont la précision est de l'ordre du mètre. Toutes ces méthodes sont complexes, nécessitent des calculs très coûteux, et sont très difficile à contrôler. Nous proposons une nouvelle approche pour générer des stalactites de glace sur des objets en fonction des paramètres de l'environnement. Cette méthode rapide permet de générer de grandes quantités de stalactites sur de grandes scènes. Le contrôle des stalactites est effectué par le biais des champs de glace qui définissent la taille des stalactites et par le biais des champs de température qui empêchent la formation des stalactites.

3.1 Définition des primitives de glace

Une primitive de glace \mathcal{P}_G est un ensemble de n stalactites candidates $\mathcal{P}_G = \{\mathcal{S}_i \mid i \in [0, n-1]\}$. Une stalactite candidate $\mathcal{S}_i = (\mathbf{p}_i, R, r_o, l)$ est caractérisée par sa position \mathbf{p}_i , sa matrice de rotation R , son rayon initial r_o et sa longueur l (Figure 4.24).

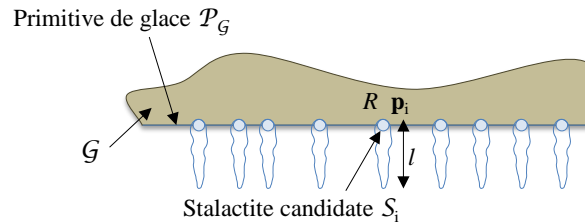


FIGURE 4.24 – Une primitive de glace regroupe un ensemble de stalactites candidates.

Nous utilisons un seul type de primitive de glace : une primitive linéaire, utilisée pour distribuer des stalactites sur un segment de droite. Ses paramètres sont les deux extrémités du segment \mathbf{p}_0 et \mathbf{p}_1 , et une distance d qui fait référence à la distance entre les positions \mathbf{p}_i de deux stalactites adjacentes. Lors de la création des stalactites candidates, nous multiplions cette distance par une fonction de bruit afin d'éviter la répétitivité. Cette primitive convient à une très

grande variété de situations : elle est utilisée pour distribuer des stalactites le long d'un banc, autour d'une tête de lampadaire, au bord et sous les vasques d'une fontaine, etc. (Figure 4.25).



FIGURE 4.25 – Les stalactites de ce lampadaire sont issues de primitives de glace linéaires.

3.2 Sélection, paramétrisation et création du modèle géométrique

L'ajout d'une stalactite dans la scène se fait en trois étapes : sélection des stalactites candidates qui vont exister, détermination de leurs paramètres, et création des modèles géométriques correspondants.

3.2.1 Sélection

L'instanciation d'une stalactite candidate est déterminée en évaluant la fonction $\mathcal{G}l$ et la fonction de température θ à sa position \mathbf{p}_i (Figure 4.26). La valeur du champ de glace $\mathcal{G}l(\mathbf{p}_i)$ doit être supérieure à zéro et la valeur du champ de température $\theta(\mathbf{p}_i)$ ne doit pas être positive. L'ensemble des stalactites candidatesinstanciées en fonction de l'environnement $glace(e)$ est donc défini comme :

$$glace(e) = \{\mathcal{S}_i \mid \mathcal{G}l(\mathbf{p}_i) > 0, \theta(\mathbf{p}_i) \leq 0\}.$$

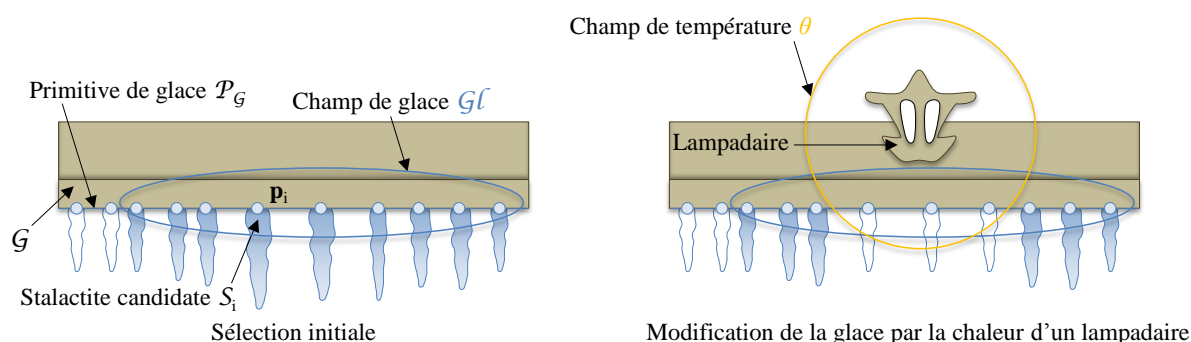


FIGURE 4.26 – L'existence d'une stalactite candidate dépend des champs de glace et des champs de température.

3.2.2 Détermination des paramètres

Une fois les stalactites candidates sélectionnées, il faut déterminer leurs paramètres : la longueur l et le rayon d'origine r_o . La longueur l d'une stalactite \mathcal{S}_i est calculée de la façon suivante : $l = 0.15 * \log(\mathcal{G}l(\mathbf{p}_i) + 1)$. Son rayon d'origine r_o est obtenu par le calcul $r_o = 0.01 * \log(\mathcal{G}l(\mathbf{p}_i) + 1)$. Ces deux valeurs sont légèrement perturbées stochastiquement pour éviter la répétition entre stalactites adjacentes. La matrice de rotation R définit une rotation aléatoire sur

l'axe vertical. Elle est utilisée uniquement dans le cas où l'on instancie des modèles géométriques pré-calculés.

3.2.3 Création du modèle géométrique

Nous utilisons pour générer le modèle géométrique des stalactites une représentation basée sur des particules et générée par un BlobTree [146]. Une liste de primitives sphériques est générée le long de l'axe verticale de la stalactite, à partir de son sommet et en descendant (Figure 4.27). Le rayon des sphères décroît, ainsi que l'espace qui les sépare. Nous appliquons également une perturbation stochastique sur la position de ces sphères afin d'introduire de la variété.

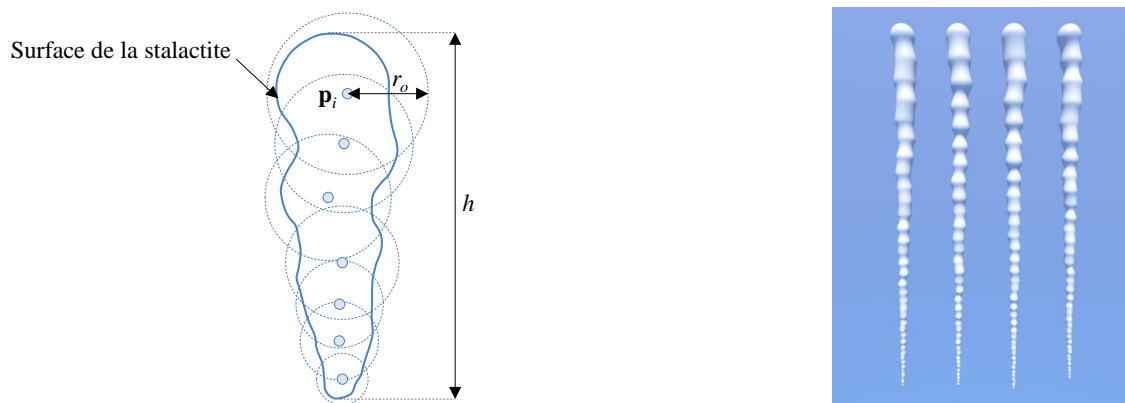


FIGURE 4.27 – Génération du modèle géométrique d'une stalactite par un ensemble de primitives sphériques du modèle BlobTree [146].

L'utilisation d'un modèle implicite pour créer la géométrie des stalactites présente un avantage important : il est très naturel d'extraire une seule surface pour plusieurs stalactites suffisamment proches, et ainsi d'obtenir un résultat de bonne qualité. La figure 4.28 présente deux stalactites \mathcal{S}_i et \mathcal{S}_j , suffisamment proches l'une de l'autre pour que les potentiels des primitives sphériques se rejoignent. Une seule surface contenant les deux stalactites est alors générée par l'algorithme d'extraction de surface.

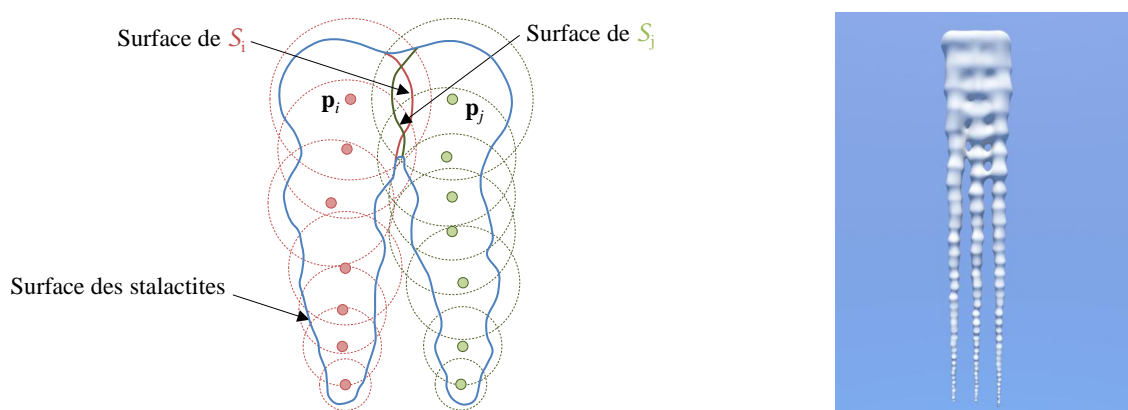


FIGURE 4.28 – Extraction d'une seule surface pour plusieurs stalactites suffisamment proches grâce aux primitives implicites.

Cependant, obtenir la surface polygonale d'un modèle implicite nécessite l'utilisation d'un algorithme d'extraction de surface tel que l'algorithme *Marching Cube* [76]. Les algorithmes de

ce type sont coûteux, ce qui implique des temps d'exécution relativement longs. Pour accélérer le calcul de la géométrie des stalactites, nous générons un ensemble réduit de modèles de stalactites lors d'une phase de pré-calcul, et nous les stockons avec leurs paramètres. Les stalactites sont ensuite instanciées à partir des modèles pré-calculés, en choisissant le modèle dont les paramètres sont les plus proches.



FIGURE 4.29 – Comparaison entre les surfaces réelles générées par le BlobTree (a) et les stalactites instanciées depuis des modèles pré-calculés (b).

La figure 4.29 compare les résultats produits par ces deux méthodes. Les temps de calculs sont respectivement de 97 secondes pour l'image 4.29(a) et de 0,001 secondes pour l'image 4.29(b). Cet écart, en plus d'être dû à la différence de fonctionnement des deux méthodes, s'explique également par le fait que l'implémentation de la méthode d'extraction de surface que nous utilisons n'est ni optimisée ni parallélisée. L'instanciation de modèles pré-calculés permet de plus de réduire considérablement la consommation mémoire. Dans l'image 4.29(a) les stalactites générées par le modèle implicite occupent 651 mégaoctets de mémoire. Au contraire, dans l'image 4.29(b), les cinq modèles de stalactites différents instanciés nécessitent un total de 3.84 mégaoctets. L'instanciation est donc parfaitement adaptée à l'édition interactive, contrairement à l'utilisation du formalisme implicite qui est à réserver au cas où l'on souhaite visualiser les stalactites de très près.

3.3 Résultats et conclusion

Les méthodes de génération de glace que nous avons analysées dans l'état de l'art (Chapitre 1) sont coûteuses, complexes à mettre en œuvre, et difficilement contrôlables. La méthode que nous proposons est rapide et facile à contrôler grâce aux champs d'influence. Les temps de calculs donnés dans les statistiques (Tableau 4.2) concernent la version optimisée de la méthode qui exploite l'instanciation de modèles pré-calculés.

Scène	Temps (s)	Instances (m)	Taille (m)
Raffinerie (Figure 4.30)	0.01	0.84	10 x 11
Parc (Figure 4.31)	0.03	14	115 x 90

Tableau 4.2 – Statistiques de la génération de glace. Le temps de génération est exprimé en secondes, le nombre d'instances en milliers et la taille des scènes en mètres.

Les 840 stalactites qui se forment sous les tuyaux de la raffinerie (Figure 4.30) sont générées en 0.01 seconde grâce à des champs de glace dont les squelettes sont des segments qui sont alignés sur l'axe des tuyaux. Tous les tuyaux de cette raffinerie sont dotés de primitives de glace, mais seuls ceux qui ne transportent pas de matériaux chauds (et qui ne génèrent donc pas de champs de chaleur) produisent des stalactites.

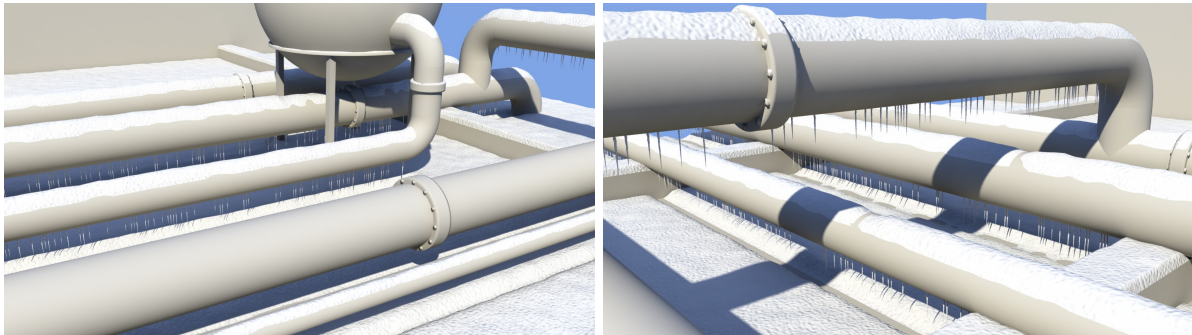


FIGURE 4.30 – Des stalactites se forment sous les tuyaux de la raffinerie qui ne transportent pas de matériaux chauds en hiver.

La fontaine et l'ensemble des lampadaires du parc génèrent un total de environ 14 000 stalactites (Figure 4.31), et seulement 0.03 secondes sont nécessaires pour calculer cette distribution.



FIGURE 4.31 – En hiver la fontaine et les lampadaires du parc sont recouverts de stalactites.

Conclusion

La nouvelle méthode de génération de glace que nous proposons pour les objets environnementaux exploite un modèle implicite pour la génération de stalactites de glace. Elle peut-être accélérée en ayant recours à l'instanciation massive de modèles pré-calculés.

Cette méthode est sujette à plusieurs limitations. Elle est conçue pour représenter de la glace uniquement sous la forme de stalactites, et pas sous d'autres types de formation de glace comme le givre que l'on peut rencontrer dans le monde réel. L'utilisation du modèle implicite permettrait cependant d'étendre les formations de glace exploitables au sein de notre système. La version non accélérée de la méthode produit des résultats de meilleure qualité, mais elle est trop lente et possède une empreinte mémoire trop lourde pour être exploitée dans le contexte de l'édition interactive.

4 Herbe

L'herbe est le troisième effet implémenté dans notre système. La distribution de l'herbe est généralement calculée grâce à des simulations d'écosystèmes comme celle présentée dans [38], mais ces approches sont très coûteuses et ne permettent pas de contrôler facilement le résultat final. Comme pour les autres effets, nous proposons une nouvelle approche pour la génération d'herbe utilisant des primitives spécifiques à cet effet. Cet effet est contrôlé à l'aide de champs d'herbe qui définissent la présence d'herbe, de champs d'accessibilité qui suppriment des touffes d'herbes, et de champs de température et d'humidité qui impactent l'état de l'herbe.

4.1 Définition des primitives d'herbe

Une *primitive d'herbe*, notée $\mathcal{P}_{\mathcal{H}}$, est une collection de n touffes d'herbes candidates $\mathcal{P}_{\mathcal{H}} = \{\mathcal{H}_i, i \in [0, n - 1]\}$. Elle contient également un atlas \mathcal{A} contenant j modèles géométriques de touffes d'herbes. Une touffe d'herbes $\mathcal{H}_i = (\mathbf{p}_i, R, H, m_i)$ est définie par sa position \mathbf{p}_i , sa matrice de rotation R , son facteur d'homothétie H et son index m_i qui fait référence à un modèle géométrique texturé (Figure 4.32). Ce modèle est choisi en fonction des champs d'herbe \mathcal{H} , de température θ et d'humidité η . Les touffes d'herbes d'une primitive sont appelées des touffes d'herbes candidates, puisque leur existence dans la scène finale dépend des paramètres de l'environnement.

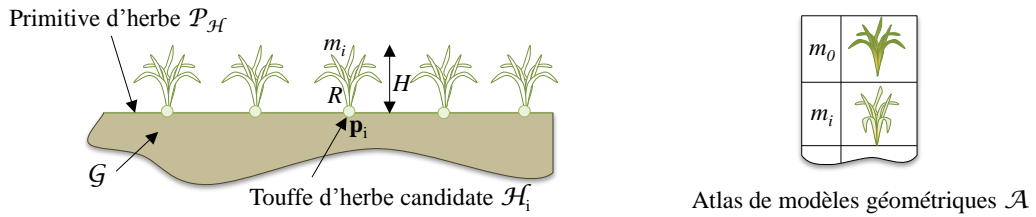


FIGURE 4.32 – Une primitive d'herbe contient un ensemble de touffes d'herbes candidates \mathcal{H}_i et un atlas de modèles géométriques de touffes d'herbes \mathcal{A} .

4.2 Les différentes primitives d'herbe

Nous avons créé plusieurs types de primitives d'herbe avec différents squelettes de support (linéaire, quadrangulaire, et à maillage quelconque) pour répondre aux différents besoins rencontrés dans les scènes que nous modélisons.

Primitive linéaire Cette primitive permet de distribuer de l'herbe sur un segment de droite. Elle est paramétrée par les deux extrémités du segment \mathbf{p}_0 et \mathbf{p}_1 et par une distance d .



FIGURE 4.33 – L’herbe sèche qui pousse entre les dalles et dans les fractures des dalles est générée à l’aide de primitives linéaires.

La distance de ce type de primitive correspond à la distance entre deux touffes d’herbes. Lors de la création des éléments candidats, cette distance est perturbée par un bruit aléatoire afin d’éviter la répétitivité. Cette primitive est par exemple utilisée pour distribuer de l’herbe autour d’une dalle, d’un trottoir, entre deux marches d’un escalier, le long d’un bâtiment, ou le long de fractures et de fissures (Figure 4.33).

Primitive quadrangulaire Cette primitive est conçue pour distribuer de l’herbe sur un polygone quadrangulaire convexe. Ses paramètres sont les quatre sommets du polygone $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ et \mathbf{v}_3 , et le rayon r . Nous l’utilisons principalement pour distribuer de l’herbe sur de grandes surfaces planes, telles que les pelouses du parc (Figure 4.34).



FIGURE 4.34 – Le sol du parc est constitué d’objets environnementaux *boites* qui sont dotés de primitives quadrangulaires.

La distribution des touffes d’herbes candidates \mathcal{H}_i est calculée à l’aide d’un pavage de tuiles créées par la méthode d’échantillonnage de Poisson. Cette méthode d’échantillonnage génère une distribution de points irrégulière, sans garantir le nombre de points distribués. Elle nécessite deux paramètres : un rayon r et un nombre d’échantillons n . Son fonctionnement est le suivant : elle tire aléatoirement un point \mathbf{p} et vérifie la distance entre \mathbf{p} et les échantillons existants. Si pour tous ces échantillons cette distance est supérieure à r , \mathbf{p} est ajouté aux échantillons existants. Dans le cas contraire, p est ignoré. Cette opération est renouvelée n fois. Voici un exemple d’implémentation de cette fonction en deux dimensions dans le carré unitaire :

```

1: function POISSONDISKS( $r, n$ )
2:    $samples := \emptyset$ 
3:   for  $i := 1; i < n; ++i$  do
4:      $x := Random(0, 1)$ 
5:      $y := Random(0, 1)$ 
6:      $p := Point(x, y)$ 
    
```

```

7:      add := true
8:      for all s in samples do
9:          if Distance(p, e) < r then
10:             add := false
11:          end if
12:        end for
13:        if add == true then
14:            samples := samples + p
15:        end if
16:      end for
17:      return samples
18: end function
    
```

Afin de créer les éléments candidats d'une primitive quadrangulaire, nous pré-calculons d'abord une tuile unitaire dans laquelle nous distribuons des échantillons avec la méthode de Poisson. Nous plongeons ensuite la primitive dans un pavage de ces tuiles et ne conservons que les échantillons qui sont situés à l'intérieur (Figure 4.35).

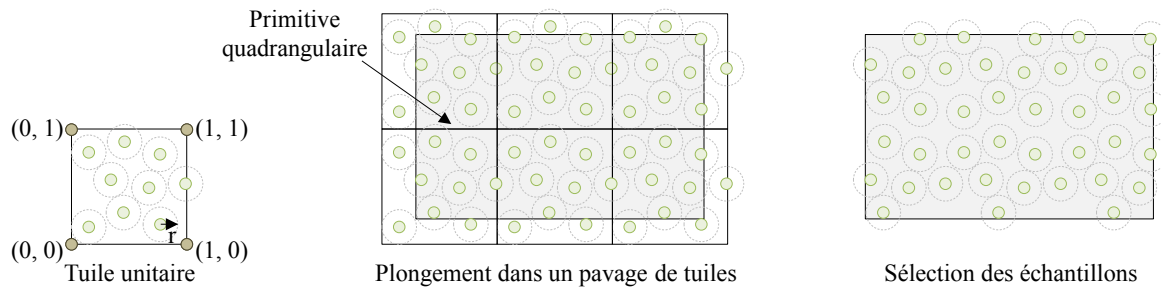


FIGURE 4.35 – La distribution des instances candidates sur une primitive quadrangulaire se fait en la plongeant dans un pavage de tuiles dont les échantillons sont distribués avec la méthode de Poisson.

Primitive à maillage quelconque Ces primitives distribuent des touffes d'herbes candidates à la surface d'un maillage quelconque. Elles nécessitent un maillage \mathcal{M} , une densité d , et une pente maximale s_{max} .

Les instances candidates sont distribuées en analysant successivement les faces f_i du maillage \mathcal{M} . Si la pente d'une face est supérieure au seuil de pente s_{max} , elle n'est pas retenue pour recevoir des instances candidates. Chaque face retenue reçoit n touffes d'herbes candidates, $n = d/A_f$ étant le rapport entre la densité d'herbe d et la surface de la face A_f . Les n touffes d'herbes candidates sont distribuées aléatoirement sur la face à laquelle elles sont liées et orientées en fonction de la normale de cette face. Ce type de primitive est utilisé pour distribuer de l'herbe sur des reliefs quelconques (Figure 4.36).

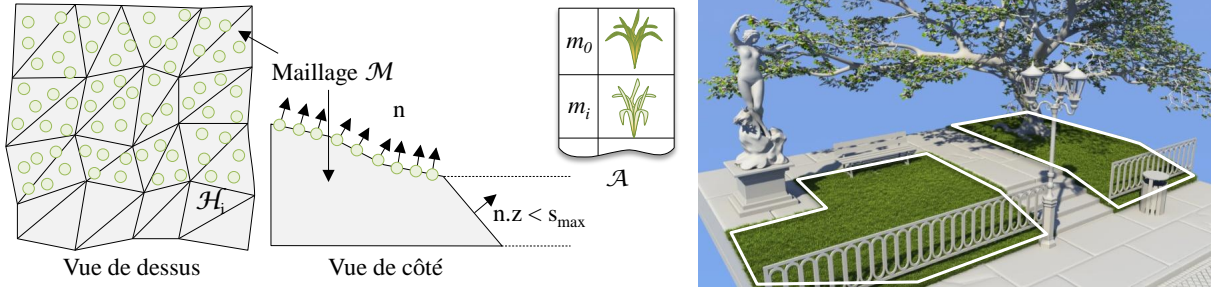


FIGURE 4.36 – Comme le sol du square est modélisé par un maillage, l’herbe est distribuée à l’aide d’une primitive à maillage.

4.3 Sélection et paramétrisation des touffes d’herbes candidates

Rappelons qu’en plus de la fonction d’herbe \mathcal{H} , les champs d’herbe définissent deux informations supplémentaires : une *densité* et un *type*. La densité $\mathcal{H}_{dens} \in [0, 1]$ d’un champ détermine la probabilité qu’une touffe d’herbes candidate située dans son aire d’influence soit effectivement instanciée, ce qui permet de contrôler la densité de touffes d’herbes se développant sur les primitives. Le type \mathcal{H}_{type} détermine le modèle géométrique m_i qui sera associé à une touffe d’herbes candidate instanciée.

4.3.1 Instanciation

L’instanciation d’une touffe d’herbes candidate \mathcal{H}_i dépend de trois paramètres : la valeur des champs d’herbe $\mathcal{H}(\mathbf{p}_i)$, l’accessibilité $\alpha(\mathbf{p}_i)$, et la densité $\mathcal{H}_{dens}(\mathbf{p}_i)$ des champs d’herbe. Trois conditions sont simultanément nécessaires pour qu’une touffe d’herbes candidate \mathcal{H}_i soit instanciée.

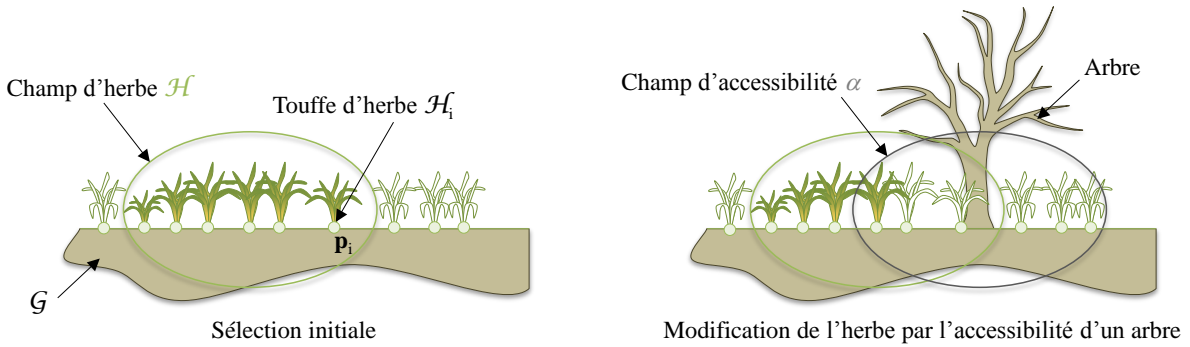


FIGURE 4.37 – L’instanciation d’une touffe d’herbes candidate dépend de la valeur des champs d’herbe et de la valeur des champs d’accessibilité.

La première condition, $\mathcal{H}(\mathbf{p}_i) > 0$, définit que la touffe d’herbes candidate \mathcal{H}_i doit être située à l’intérieur d’un champ d’herbe. La seconde condition, $\alpha(\mathbf{p}_i) > a_\alpha$, avec a_α un nombre aléatoire tel que $a_\alpha \in [0, 1]$, correspond au rôle joué par l’accessibilité dans l’instanciation d’une touffe d’herbes candidate : l’accessibilité diminue stochastiquement la probabilité qu’une touffe d’herbes candidate soit instanciée. Plus l’accessibilité de \mathbf{p}_i est faible, plus la probabilité que \mathcal{H}_i soit instanciée diminue (Figure 4.37). La troisième condition $\mathcal{H}_{dens}(\mathbf{p}_i) > a_d$, avec a_d un nombre aléatoire tel que $a_d \in [0, 1]$, permet le contrôle de la densité des touffes d’herbes instanciées en attribuant à chaque touffe d’herbes candidate une probabilité d’être instanciée (Figure 4.38).

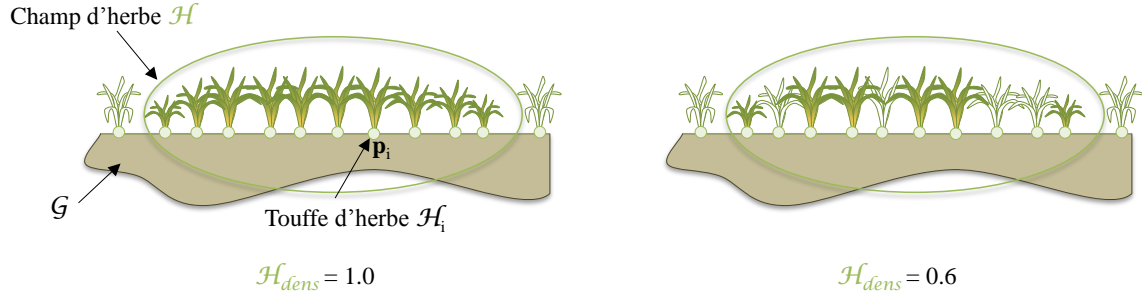


FIGURE 4.38 – L’instanciation d’une touffe d’herbes candidate dépend également du paramètre \mathcal{H}_{dens} des champs d’herbe.

La figure 4.39 montre l’impact de ce paramètre : les trois images de cette figure sont générées avec des conditions identiques, à l’exception de la densité \mathcal{H}_{dens} du champ d’herbe qui varie et vaut respectivement 1.0, 0.5, et 0.1 pour les images de gauche à droite.

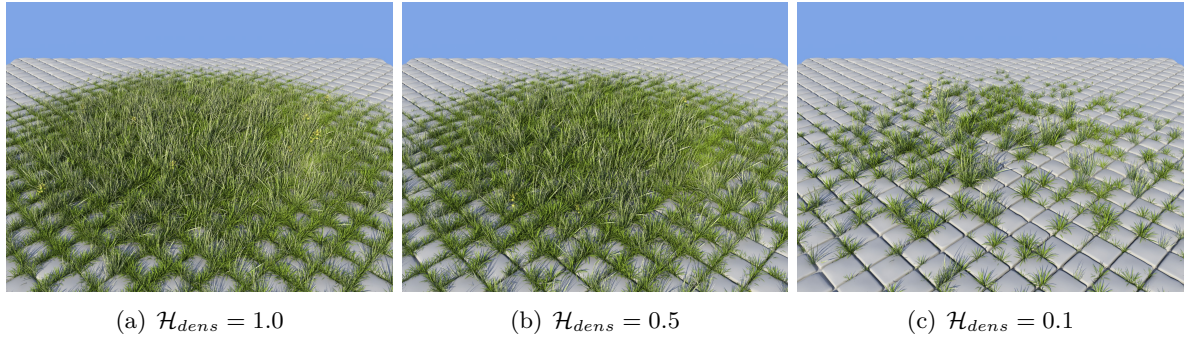


FIGURE 4.39 – La densité du champ d’herbe \mathcal{H}_{dens} est le seul paramètre qui varie entre ces trois images.

4.3.2 Détermination des paramètres

Une fois qu’une touffe d’herbes candidate \mathcal{H}_i est sélectionnée pour être instanciée, il est nécessaire de déterminer ses différents paramètres d’instanciation : sa matrice de rotation R , son facteur d’homothétie H et l’index définissant le modèle géométrique à instancier m_i .

La matrice de rotation R est simplement définie par une rotation aléatoire sur l’axe vertical dans le but de briser la répétitivité pour un coût minimal. Le facteur d’homothétie H est fourni par l’intensité $\mathcal{H}(\mathbf{p}_i)$ des champs d’herbe au point \mathbf{p}_i , multipliée par un facteur stochastique a : $h_i = a \mathcal{H}(\mathbf{p}_i)$, $a \in [0.1, 1.5]$. Ce facteur permet d’introduire de la variété dans la taille des touffes d’herbes adjacentes, afin de ne pas être soumis à un résultat anormalement lissé qui serait produit en utilisant uniquement l’intensité des champs d’herbe.

L’index du modèle géométrique m_i est défini en deux étapes : le calcul d’un type d’herbe suivi du calcul de son état qui est soit normal, soit desséché (Figure 4.40).



FIGURE 4.40 – Les quatre types de touffes d’herbe, en version normale à gauche et en version desséchée à droite.

Le type de la touffe d’herbes à instancier est fourni par une requête spécifique des champs d’herbe, $\mathcal{H}_{type}(\mathbf{p}_i)$, qui retourne la probabilité qu’à \mathcal{H}_i d’appartenir à chacun des quatre types d’herbes. Dans les trois images de la figure 4.41, seules les probabilités du champ d’herbe, qui sont indiquées dans le tableau, 4.3 ont été modifiées.

Image	Type 1	Type 2	Type 3	Type 4
Figure 4.41(a)	1.00	0.00	0.00	0.00
Figure 4.41(b)	0.80	0.16	0.02	0.02
Figure 4.41(c)	0.25	0.25	0.25	0.25

Tableau 4.3 – Probabilités du champ d’herbe des différentes images de la figure 4.41.

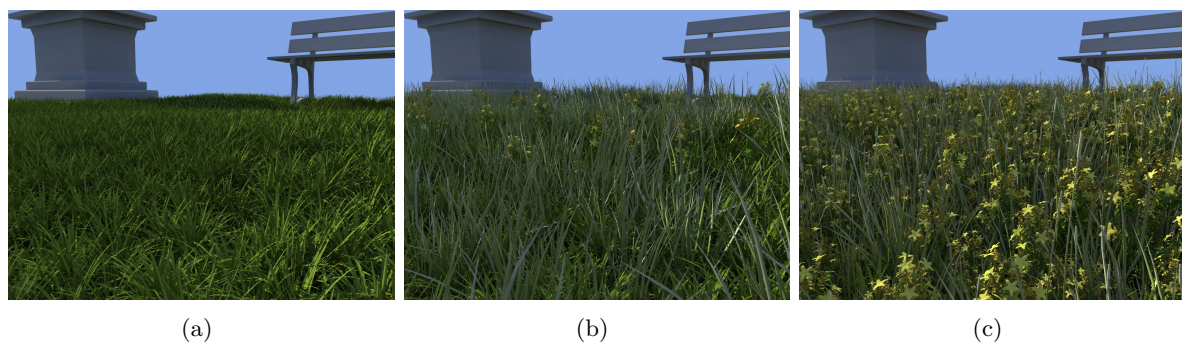


FIGURE 4.41 – Les variations entre les distributions d’herbes de ces trois images sont obtenues en modifiant les probabilités du champ d’herbe.

Les champs de température θ et les champs d’humidité η interviennent également dans le calcul du modèle géométrique des touffes d’herbe (Figure 4.42).

- Lorsqu’une touffe d’herbes candidate \mathcal{H}_i n’est influencée ni par un champ de température ni par un champ d’humidité, donc quand $\theta(\mathbf{p}_i) = 0$ et $\eta(\mathbf{p}_i) = 0$, m_i fait référence à son modèle normal.
- Lorsque \mathcal{H}_i est uniquement influencée par un champ de température, donc quand $\theta(\mathbf{p}_i) > 0$ et $\eta(\mathbf{p}_i) = 0$, m_i correspond au modèle desséché.
- Au contraire, lorsque \mathcal{H}_i est influencée à la fois par un champ de température et par un champ d’humidité, donc quand $\theta(\mathbf{p}_i) > 0$ et $\eta(\mathbf{p}_i) > 0$, m_i renvoie au modèle normal (Figure 4.43).

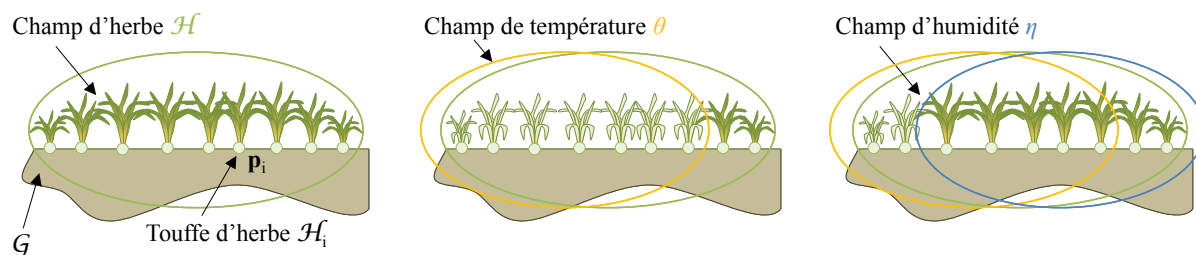


FIGURE 4.42 – Les champs de température et les champs d’humidité influent sur la sélection du modèle géométrique des touffes d’herbes instanciées.

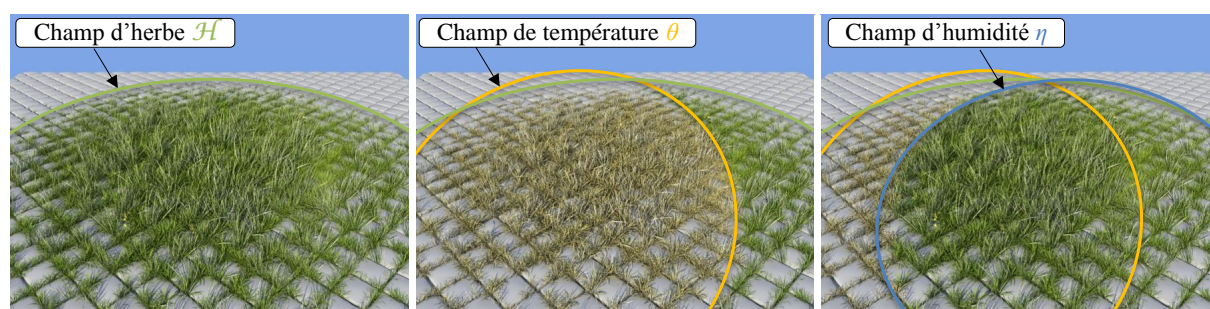


FIGURE 4.43 – Influence des champs de température et d’humidité sur le choix du modèle géométrique. Les champs de température créent des zones d’herbes sèches et les champs d’humidité annulent leur effet.

4.4 Résultats et conclusion

Les méthodes que nous avons présentées dans le chapitre 1 sont centrées soit sur le rendu d’herbe en temps réel, soit sur la distribution d’herbe en utilisant un modèle d’écosystème. Les méthodes de distribution sont coûteuses en calcul et sont peu contrôlables. Avec cette nouvelle approche, nous solutionnons ces deux problèmes. L’utilisation des champs d’influence permet de contrôler facilement le résultat et de mélanger facilement plusieurs états pour plusieurs types d’herbe. Notre méthode est peu coûteuse : le temps de génération est de l’ordre de la milliseconde pour un mètre carré d’herbe (Tableau 4.4).

Scène	Temps (s)	Instances (m)	Taille (m)
Parc (Figure 4.45)	5.01	525	115 x 90
Square (Figure 4.44)	0.10	30	10 x 10

Tableau 4.4 – Statistiques de la génération d’herbe. Le temps de génération est exprimé en secondes, le nombre d’instances en milliers et la taille des scènes en mètres.

L’herbe du parc de la figure 4.44 est générée par notre système en seulement 0.1 seconde. L’herbe séchée qui pousse sur le trottoir est le résultat de la combinaison d’un champ d’herbe et d’un champ de température. Sous le banc et au niveau des racines de l’arbre, la quantité d’herbe est diminuée. Ce résultat est obtenu en utilisant des champs d’occlusions.



FIGURE 4.44 – Le même square que dans la figure 4.36, cette fois ci dans une version décrépie.

La figure 4.45 présente une version automnale mal entretenue du parc où plusieurs types d’herbe ont colonisé les pelouses de manière désordonnée. La réduction de la quantité d’herbe au niveau des arbres est créée à l’aide de champs d’accessibilité. Cette scène nécessite 5.01 secondes de calcul, ce temps étant dû au nombre important de volumes qui sont utilisés.



FIGURE 4.45 – Une version automnale du parc. Des feuilles jonchent les pelouses à l’herbe irrégulière.

Conclusion

Nous avons proposé une méthode de génération d’herbe pour nos objets environnementaux en nous reposant sur l’instanciation massive de modèles géométriques. Les avantages de cette approche sont de permettre d’adapter la distribution d’herbe à chaque type d’objets dans la scène, d’être facilement contrôlable et très rapide.

Cette méthode est cependant limitée par différents aspects. Sur une petite scène d’une centaine de mètres carrés comme celle du square (Figure 4.44), la distribution d’herbe créée est composée de dizaines de milliers de touffes d’herbes. Même en utilisant des modèles de touffes d’herbes simples comprenant peu de polygones, la quantité totale de polygones d’herbe rend notre méthode inadaptée au rendu temps réel pour de grandes scènes. Les modèles d’écosystèmes simulent la compétition entre les espèces de végétaux et produisent des résultats crédibles, puisque leur fonctionnement interne utilise des règles et des paramètres issus de la réalité. Notre méthode axée sur le contrôle utilisateur peut nécessiter le placement de nombreux champs de contrôle pour produire des résultats plausibles. Les champs de température et d’humidité sont utilisés d’une manière directe pour déterminer le modèle géométrique d’une touffe d’herbes candidate, et améliorer la façon dont ces champs sont pris en compte permettrait d’obtenir des résultats plus naturels et de meilleure qualité.

5 Feuilles

Bien qu'il existe différents travaux sur la modélisation [93] et le vieillissement [66], [107] des feuilles, la simulation de l'accumulation de tas de feuilles en automne a été peu abordée. La méthode proposée par Desbenoit [36] qui simule le déplacement des feuilles lors de leur chute est complexe et ne permet pas de distribuer de grandes quantités de feuilles, ni de distribuer des feuilles sur de grandes surfaces. Dans cette partie nous présentons une nouvelle méthode de distribution procédurale de feuilles très rapide qui fonctionne en deux étapes et permet de résoudre ces problèmes. Nous construisons une collection de feuilles candidates sous la forme de *primitives de feuilles*, ce qui définit un ensemble d'instances virtuelles de feuilles. La distribution finale est obtenue en instanciant et en paramétrant une partie des feuilles candidates en fonction de la valeur des champs de feuilles et des champs d'accessibilité.

5.1 Définition des primitives de feuilles

Une *primitive de feuilles* est constituée d'une collection de n feuilles candidates et notée $\mathcal{P}_{\mathcal{F}} = \{\mathcal{F}_i, i \in [0, n - 1]\}$ et d'un atlas de j modèles géométriques de feuilles \mathcal{A} . Une feuille candidate $\mathcal{F}_i = \{\mathbf{p}_i, \mathbf{R}, d, m_i\}$ est définie par sa position \mathbf{p}_i , sa matrice de rotation \mathbf{R} , sa distance à la primitive d , et son index m_i qui fait référence à un modèle géométrique texturé de l'atlas \mathcal{A} (Figure 4.46).

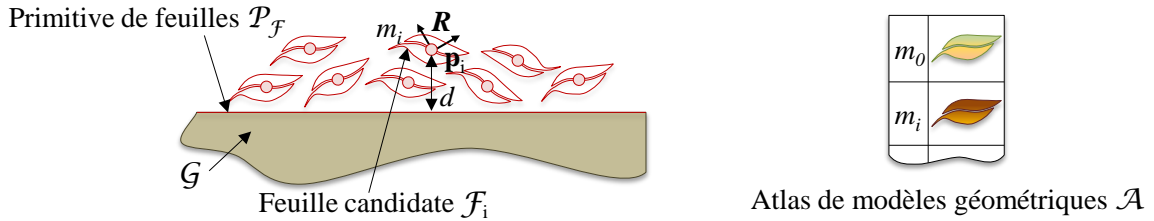


FIGURE 4.46 – Une primitive de feuilles regroupe un ensemble de feuilles candidates \mathcal{F}_i et un atlas de modèles géométriques de feuilles \mathcal{A} .

5.2 Les différentes primitives de feuilles

Nous définissons quatre types de primitives pour recouvrir les différents objets environnementaux de feuilles : les primitives à squelette quadrangulaire, à squelette discoïde, de terrain, et à nuage de points. Les trois premiers types de primitives sont conçus pour distribuer de grandes quantités de feuilles empilées en couches successives. À l'opposé, les primitives à nuage de points permettent l'instanciation de feuilles spécifiques pour correspondre à des cas particuliers et complexes.

Squelette quadrangulaire Ces primitives permettent de distribuer des feuilles sur un polygone quadrangulaire convexe. Les paramètres nécessaires à leur construction sont les sommets $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ et \mathbf{v}_3 du polygone. Ces primitives sont utilisées pour distribuer des feuilles sur des surfaces planes, tel qu'un rebord de fenêtre, une table, un banc, ou le sol (Figure 4.47).

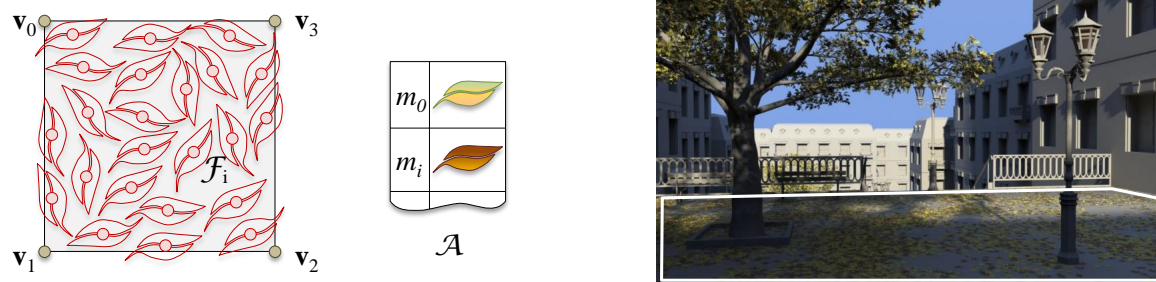


FIGURE 4.47 – Le sol de cette place est modélisé par des objets environnementaux *boite* qui comportent des primitives de feuilles quadrangulaires.

Squelette discoïde Ce type de primitive est utilisé pour distribuer des feuilles à la surface d'un disque. Les primitives discoïdes sont paramétrées par le centre du disque \mathbf{c} et par son rayon r . Nous les utilisons principalement pour distribuer des feuilles sur des éléments de mobilier urbain (Figure 4.48).

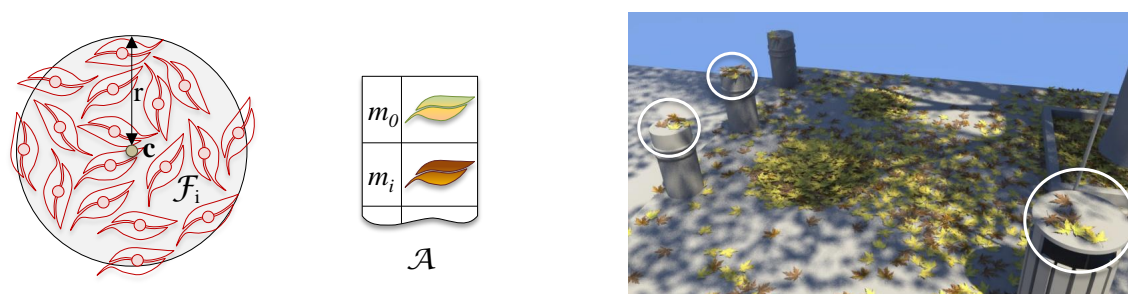


FIGURE 4.48 – Les feuilles qui se trouvent sur la poubelle et sur les plots sont générées par des primitives discoïdes.

Terrain Cette primitive permet de distribuer des feuilles sur un terrain défini par une carte d'élévation $z = f(x, y)$ (Figure 4.49). Dans notre système, le modèle de terrain que nous utilisons est issu d'un arbre de construction [108]. Les données nécessaires à cette primitive sont un terrain \mathcal{T} et les quatre sommets $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ et \mathbf{v}_3 définissant un quadrangle convexe dans lequel est inscrite la projection du terrain sur le plan $z = 0$.

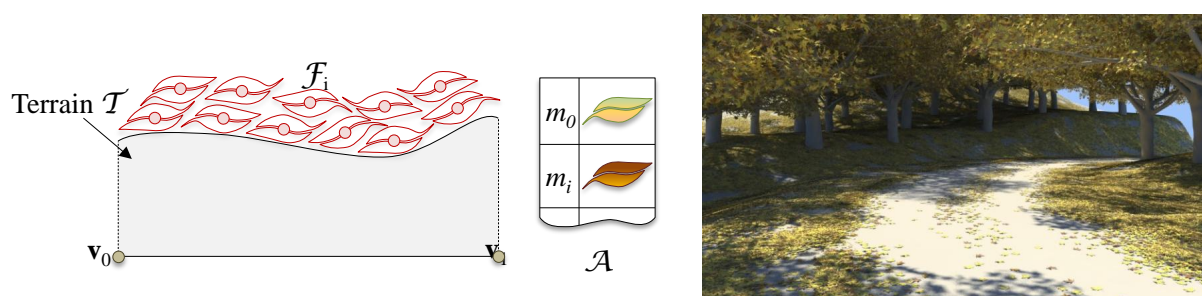


FIGURE 4.49 – Le sol de cette scène est défini par le modèle de terrain de [108] et les feuilles y sont distribuées à l'aide d'une primitive de terrain.

Nuage de points Au lieu de permettre la distribution automatique de grandes quantités de feuilles, le but de cette primitive est de permettre l'instanciation ponctuelle de feuilles dans

des cas particuliers ou complexes. Dans le monde réel, il est courant de rencontrer des feuilles accrochées et coincées dans des grillages, des barrières ou des filets (Figure 4.50). Ces cas sont extrêmement difficiles à reproduire avec des simulations physiques de vent et des détections de collisions à cause de la quantité et de la complexité des calculs nécessaires.

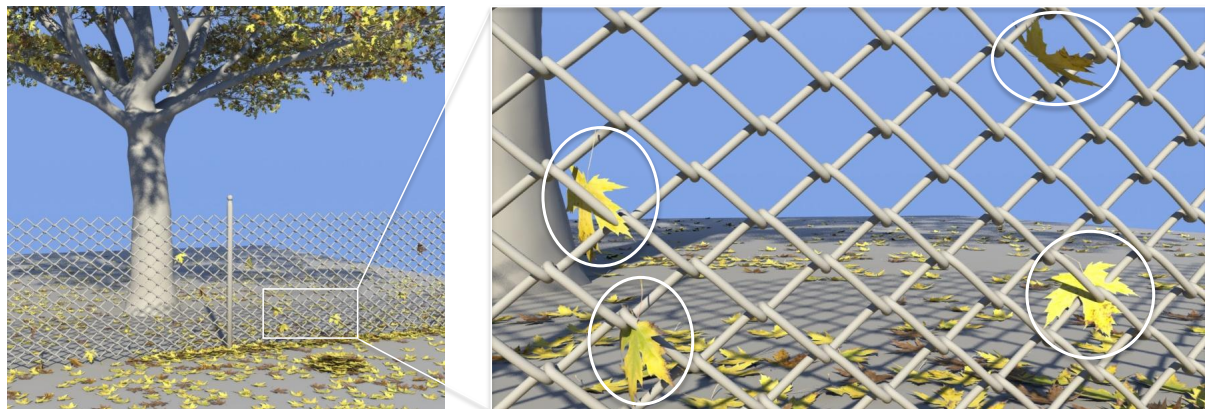


FIGURE 4.50 – Les feuilles au sol sont distribuées par des primitives quadrangulaires alors que celles des branches de l'arbre ainsi que celles accrochées au grillage sont créées grâce à des primitives à nuage de points.

Au lieu d'être calculés automatiquement, les paramètres des feuilles candidates des primitives à nuage de points sont édités par l'utilisateur. Cette étape est obligatoire pour contrôler précisément la distribution. De cette façon, il est facile d'ajouter des feuilles candidates à des emplacements spécifiques. Les feuilles étant accrochées à la géométrie \mathcal{G} de l'objet, la distance attribuée à chacune de ces feuilles candidates est $d = 0$, ce qui permet d'utiliser le même algorithme que pour les autres types de primitives lors de la sélection des feuilles à instancier (Section 5.4).

5.3 Distribution des feuilles candidates

Pour les primitives à squelette quadrangulaire et discoïde ainsi que pour les primitives de terrain, la construction de la distribution des feuilles candidates est faite lors d'une étape de pré-calcul qui définit les positions \mathbf{p}_i et les distances d . Nous utilisons pour ce faire une approche de pavage apériodique en trois dimensions inspirée de l'algorithme de génération de tas de cailloux présenté dans [109].

La méthode qui attribue des feuilles candidates à ces primitives fonctionne en deux étapes. Nous générons d'abord une tuile de feuilles cubique en superposant des couches de feuilles. Ces couches contiennent des feuilles qui pavent l'espace de façon apériodique (Figure 4.51). Les positions \mathbf{p}_i des feuilles candidates de chaque couche sont créées en utilisant une distribution de Poisson.

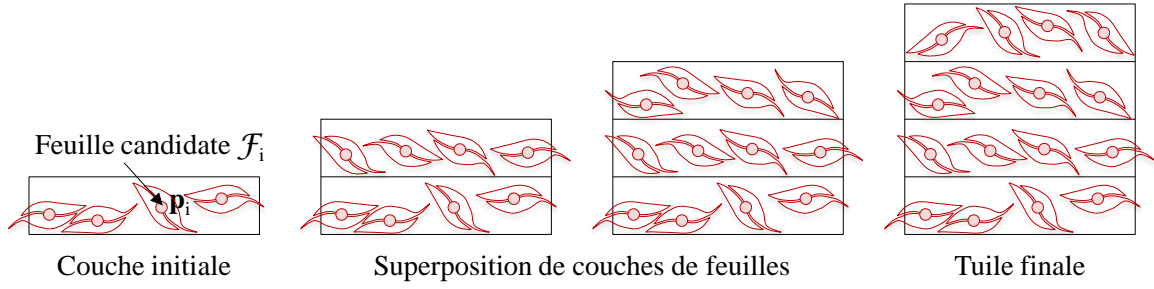


FIGURE 4.51 – Empilement de couches de feuilles distribuées par la méthode de Poisson pour générer une tuile cubique.

Dans le cas des primitives à squelette quadrangulaire et discoïde, nous plongeons la primitive $\mathcal{P}_{\mathcal{F}}$ dans un pavage de tuiles de feuilles. Les feuilles candidates de $\mathcal{P}_{\mathcal{F}}$ sont celles qui pourraient reposer sur la primitive, donc celles situées à sa verticale (Figure 4.52).

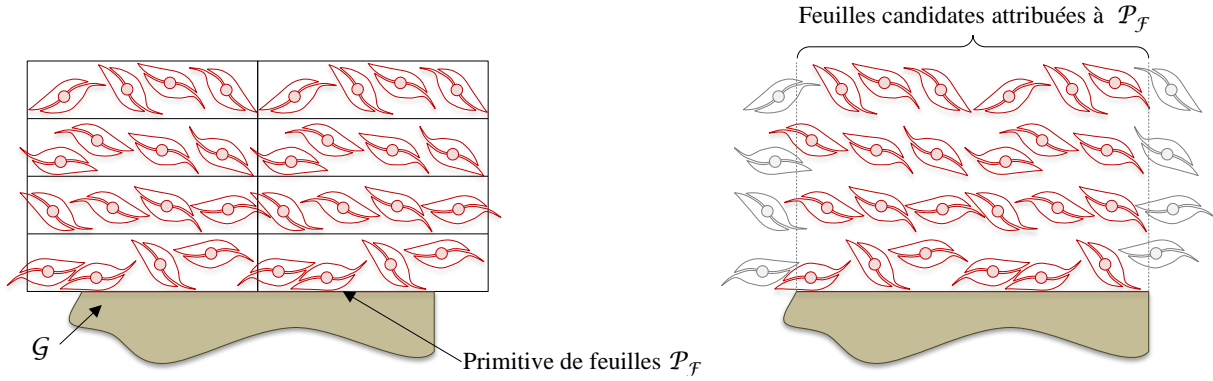


FIGURE 4.52 – L'attribution des feuilles candidates à une primitive se fait en plongeant la primitive dans un pavage de tuile.

Le processus de distribution des feuilles candidates est plus complexe dans le cas des primitives de terrain (Figure 4.53). Nous créons d'abord une primitive quadrangulaire correspondant à la projection du terrain sur le plan $z = 0$, et nous distribuons des feuilles candidates sur cette primitive. Notre modèle de terrain définit une fonction continue $z(\mathbf{p})$ qui retourne l'altitude de n'importe quel point \mathbf{p} du terrain en fonction de son abscisse et de son ordonnée. Nous déplaçons finalement les instances candidates en fonction de la hauteur du terrain : leur nouvelle position devient $\mathbf{p}_i = \mathbf{p}_i + (0, 0, z(\mathbf{p}_i))$. Nous orientons également la matrice de rotation \mathbf{R} des feuilles candidates en fonction de la normale de $z(\mathbf{p}_i)$ afin que les feuilles soient parallèles au terrain.

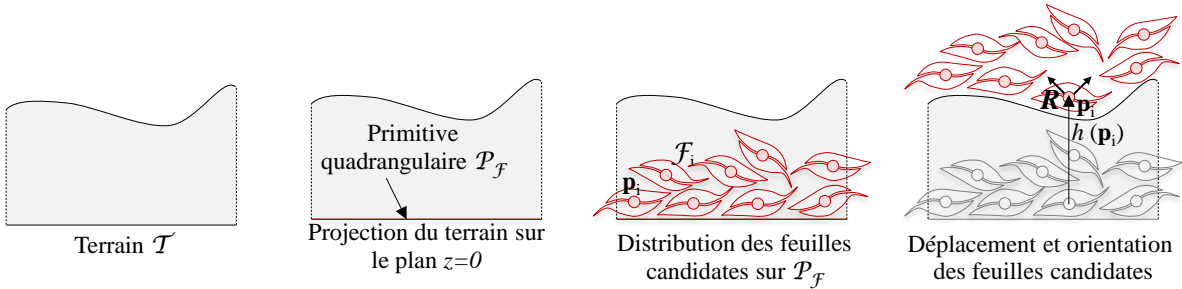


FIGURE 4.53 – Les feuilles candidates d'une primitive de terrain sont d'abord distribuées sur la projection du terrain sur le plan $z = 0$ puis déplacées en fonction de la hauteur du terrain.

Une fois que les feuilles candidates sont distribuées à la surface de la primitive, l'étape suivante consiste à sélectionner et à instancier les feuilles candidates en fonction de l'environnement.

5.4 Sélection et paramétrisation des feuilles candidates

Les champs de feuilles déterminent non seulement la fonction de feuilles \mathcal{F} , mais également une information supplémentaire : le *type* des feuilles, qui détermine le modèle géométrique correspondant à une feuille instanciée.

5.4.1 Sélection

L'instanciation d'une feuille candidate \mathcal{F}_i dépend de la valeur $\mathcal{F}(\mathbf{p}_i)$ de la fonction de feuilles, de la valeur $\alpha(\mathbf{p}_i)$ de la fonction d'accessibilité et de sa distance d à la primitive (Figure 4.54). Les champs d'accessibilité sont pris en compte comme un facteur multiplicatif, afin que les objets proches puissent empêcher l'accumulation de feuilles. \mathcal{F}_i est instanciée si la valeur de la fonction de feuilles multipliée par la valeur de la fonction d'accessibilité est supérieure à la distance de \mathcal{F}_i à la primitive $\mathcal{P}_{\mathcal{F}}$ à laquelle elle est associée, donc si $\alpha(\mathbf{p}_i) \mathcal{F}(\mathbf{p}_i) > d$. Afin de briser la régularité des couches de feuilles, nous utilisons une fonction de bruit b pour perturber l'instanciation des feuilles candidates : au lieu de prendre en compte uniquement la distance d , nous considérons $d * b(\mathbf{p}_i)$. De cette façon, la distance d de certaines des feuilles est artificiellement augmentée, ce qui diminue localement la quantité de feuilles sélectionnées.

L'ensemble des feuilles candidates instanciées en fonction de l'environnement $feuilles(e)$ est donc défini comme :

$$feuilles(e) = \{\mathcal{F}_i \mid \alpha(\mathbf{p}_i) \mathcal{F}(\mathbf{p}_i) > (d * b(\mathbf{p}_i))\}.$$

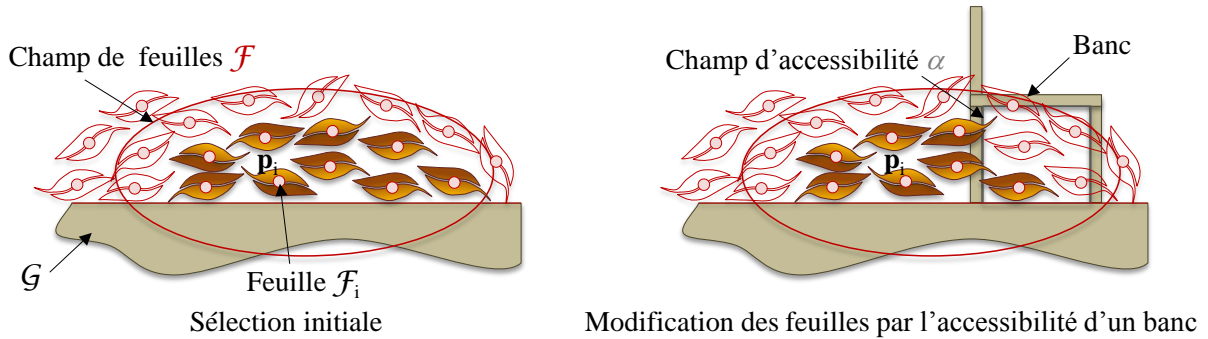


FIGURE 4.54 – L'instanciation d'une feuille candidate est liée à la valeur des champs d'herbe et à la valeur des champs d'occlusion.

5.4.2 Détermination des paramètres

Après la sélection des feuilles candidates à instancier, il est nécessaire de déterminer leurs paramètres : la matrice de rotation R et le modèle géométrique m_i . La matrice de rotation R définit une rotation aléatoire sur l'axe verticale pour éviter les répétitions.



FIGURE 4.55 – Nous utilisons cinq modèles de feuilles de platane (a) et neuf modèles de feuilles de cerisiers du japon (b).

Le modèle géométrique m_i des feuilles instanciées est fourni par la requête \mathcal{F}_{type} , qui est propre aux champs de feuilles. Nous utilisons deux espèces d'arbre, des platanes et des cerisiers du Japon, pour lesquels nous disposons respectivement de cinq et neuf modèles de feuilles (Figure 4.55). Pour une feuille candidate \mathcal{F}_i qui doit être instanciée, $\mathcal{F}_{type}(\mathbf{p}_i)$ détermine stochastiquement son modèle en prenant en compte la probabilité qu'elle a d'appartenir à chacun des quatorze types (Figure 4.56).

Image	Type 1	Type 2	Type 3	Type 4	Type 5
Figure 4.56(a)	0.00	0.00	0.50	0.50	0.00
Figure 4.56(b)	0.50	0.10	0.20	0.00	0.20
Figure 4.56(c)	0.10	0.45	0.00	0.00	0.45

Tableau 4.5 – Probabilités des champs de feuilles des différentes images de la figure 4.56. Comme la scène de cette figure contient uniquement un platane, seules les probabilités des feuilles de platanes sont détaillées.

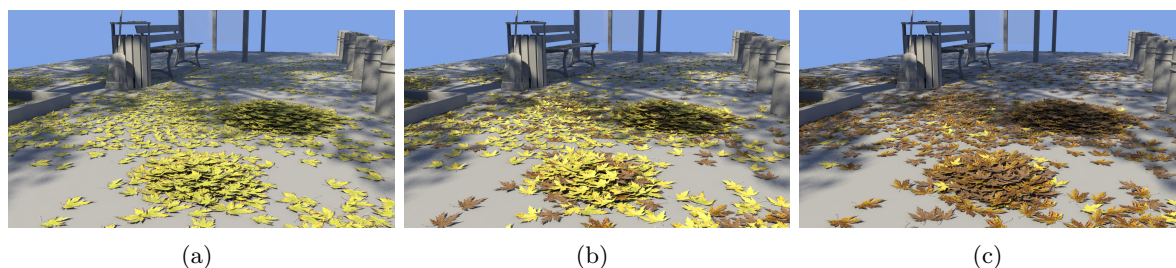


FIGURE 4.56 – Le seul paramètre qui varie entre ces trois images est la probabilité des champs de feuilles. Elles sont détaillées dans le tableau 4.5.

5.5 Résultats et conclusion

Notre nouvelle méthode de distribution de feuilles produit des résultats supérieurs d'un ordre de grandeur à ceux existant dans l'état de l'art : elle permet de distribuer plusieurs centaines de milliers de feuilles sur des surface de plusieurs milliers de mètres carré en quelques secondes seulement (Tableau 4.6). Les champs d'influence permettent de contrôler facilement la distribution de milliers de feuilles de différents types, et les primitives à nuage de points offrent

un contrôle extrêmement précis et rendent possible la création de caractéristiques de haut niveau telles que des feuilles accrochées dans du grillage.

Scène	Temps (s)	Instances (m)	Taille (m)
Montmartre (Figure 4.57)	0.54	221	33 x 100
Parc (Figure 4.58)	4.53	839	115 x 90
Route (Figure 4.59(b))	6.73	1 416	70 x 70

Tableau 4.6 – Statistiques de la génération de feuilles. Le temps de génération est exprimé en secondes, le nombre d’instances en milliers et la taille des scènes en mètres.

La figure 4.57 présente le quartier similaire à Montmartre dans une version automnale. Il comprend environ 221 000 feuilles qui sont générées par les champs d’influence des 4 platanes situés sur trois niveaux différents. Ces feuilles sont distribuées en 0.54 secondes.



FIGURE 4.57 – Le quartier au style architectural proche de Montmartre en automne.

Le parc de la figure 4.58 contient 18 platanes dont les champs d’influence génèrent environ 839 000 feuilles en 4.53 secondes.



FIGURE 4.58 – Comme le parc contient un nombre important d’arbres il est recouvert d’une grande quantité de feuilles en automne.

Nous avons produit d’autres scènes très variées pour montrer la versatilité de notre système, adapté aussi bien à la création de petites scènes comportant quelques centaines de feuilles qu’à la création de grandes scènes qui en comportent plusieurs centaines de milliers (Figure 4.59). La route de montagne est la scène qui contient le plus de feuilles parmi celles que nous avons présentées : environ 1 416 000 feuilles y sont distribuées en 6.73 secondes.



FIGURE 4.59 – Cette entrée de ce domaine et cette route de montagne sont des exemples de scènes variées que l’on peut créer avec notre système.

Conclusion

La méthode que nous proposons pour générer des feuilles sur les objets environnementaux est basée sur l’instanciation massive de modèles géométriques pavés aperiodiquement. Elle permet de distribuer de grandes quantités de feuilles ou de contrôler très précisément leur distribution. La méthode de pavage utilisée est généralisable à un grand nombre d’autres objets, et nous l’utilisons pour la génération de pétales et d’ordures (Figure 4.60). Elle pourrait également être utilisée pour la distribution de brindilles, d’écorce, de cailloux, d’un mélange de galets et de coquillages, etc.

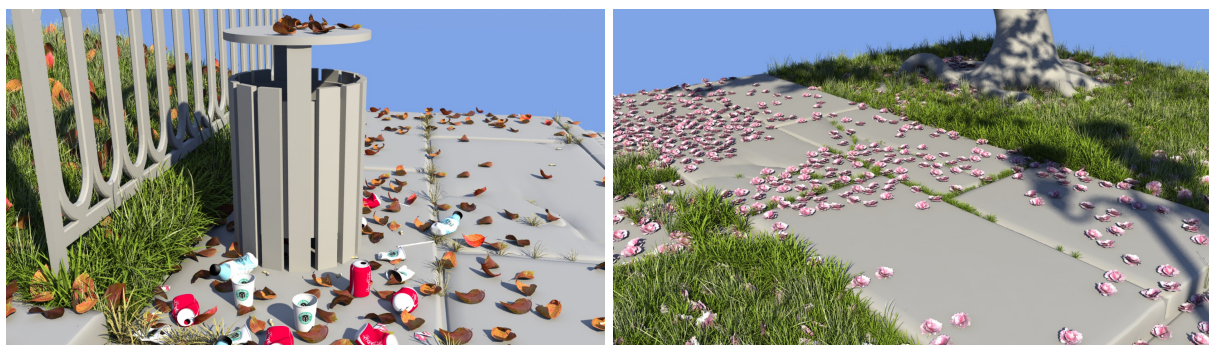


FIGURE 4.60 – La méthode de distribution des feuilles peut être utilisée pour distribuer d’autres types d’objets comme des déchets ou des pétales.

Cette méthode est sujette à plusieurs limitations. L’instanciation massive de modèle géométrique la rend difficilement exploitable pour de la visualisation en temps réel. Les primitives à nuage de points offrent un contrôle extrêmement précis sur la distribution de feuilles, mais elles requièrent une intervention de l’utilisateur qui peut devenir fastidieuse si le nombre de feuilles candidates à éditer est important. Les primitives de terrains ne prennent pas en compte la pente lors de la distribution, ce qui peut produire des dépôts de feuilles dans des zones où la pente les rendrait théoriquement impossibles.

6 Accélération

De multiples occurrences d'un objet environnemental coexistent très souvent au sein d'une scène. À l'origine, lors de l'évaluation de la scène, les effets de ce type d'objets sont calculés pour chacune des occurrences de chaque type d'objet environnemental. Dans une scène qui comporte un grand nombre d'objets environnementaux identiques, tel qu'un immeuble qui comprend de nombreuses fenêtres, cela peut représenter une partie importante du temps de calcul nécessaire à l'évaluation totale de la scène. Les effets des objets environnementaux qui ont les mêmes paramètres environnementaux vont être quasi identiques, aux variations aléatoires (bruit qui modifie l'élévation de la neige, distance entre deux stalactites adjacentes, etc.) près.

Pour accélérer l'évaluation de la scène notre système peut détecter, en traversant l'arbre de construction de la scène, les occurrences d'un même type d'objets environnementaux qui ont les mêmes paramètres environnementaux. La génération des effets est alors appelée une seule fois pour tous ces objets, et l'effet résultant est ensuite dupliqué pour chaque occurrence de cet objet environnemental. La figure 4.61 montre une scène comportant un lampadaire et un immeuble. Le lampadaire émet un champ de température θ , et les deux fenêtres dont le fond est rouge sont influencées par ce champ. Elles vont donc calculer chacune un effet de neige qui leur sera spécifique. Au contraire, toutes les fenêtres dont le fond est blanc partagent les mêmes paramètres environnementaux, étant donné qu'elles ne sont sujettes à aucun champ d'influence. Lors de l'évaluation de la scène, la première des fenêtres à fond blanc rencontrée lors de la traversée de l'arbre va générer un maillage de neige, et ce maillage sera ensuite instancié pour toutes les fenêtres identiques.

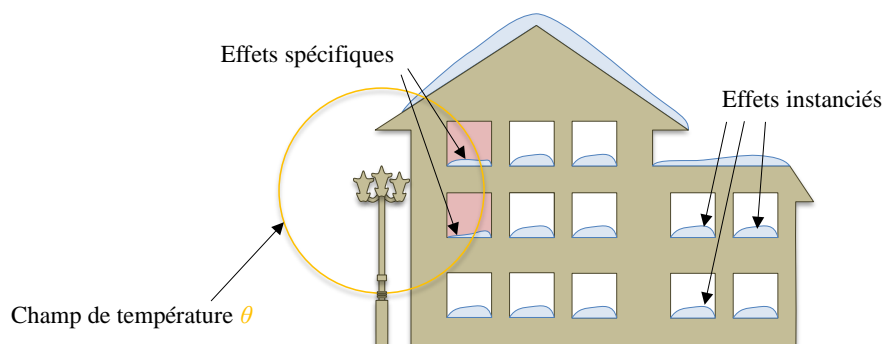


FIGURE 4.61 – Le processus de calcul des effets de neige n'est appelé qu'une fois pour toutes les fenêtres dont le fond est blanc. Au contraire, les fenêtres dont le fond est rouge génèrent chacune leur propre effet de neige.

Un exemple de fenêtres et de balcons instanciés sur un building créé par une grammaire de forme enrichie est montré dans la figure 4.62. Le bâtiment possède au total 70 fenêtres et balcons, ainsi que 4 lampadaires sur sa façade. Les 70 fenêtres et balcons sont instanciés, et leurs effets n'ont besoin d'être calculés que 6 fois : une fois pour chacune des 5 fenêtres du bas de la façade qui sont sous l'influence du champ de température produit par les lampadaires, et seulement une autre fois pour l'ensemble des autres fenêtres. Dans la version non accélérée, 0.44 secondes sont nécessaires au calcul de la neige de ce bâtiment. Une fois les fenêtres instanciées, ce temps passe à 0.28 secondes.

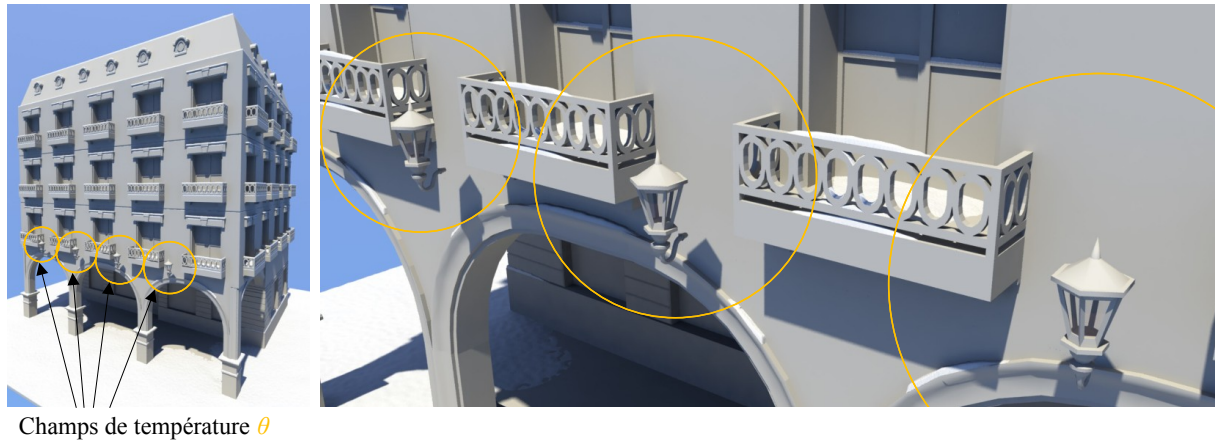


FIGURE 4.62 – Un bâtiment dont les fenêtres sont instanciées. Les fenêtres situées dans les champs de chaleur sont les seules pour lesquelles l’effet de neige a besoin d’être calculé spécifiquement. Les décorations de neige sont identiques pour toutes les autres fenêtres.

7 Conclusion

Nous avons proposé dans ce chapitre de nouvelles méthodes qui offrent plusieurs avantages pour la génération des effets de neige, de glace, de feuilles et d’herbe pour les objets environnementaux. Ces méthodes reposent sur l’assignation aux objets environnementaux de primitives spécifiques à chaque effet. Les primitives sont des supports qui génèrent des maillages ou des instances de modèles géométriques dont l’existence et les paramètres sont déterminés par les paramètres de l’environnement. Nos méthodes offrent les avantages suivants.

Échelle Elles permettent de générer de très nombreux détails à la précision inférieure au centimètre dans de grandes scènes de plusieurs milliers de mètres carrés, offrant ainsi un facteur d’échelle au moins supérieur d’un ordre de grandeur à ceux des méthodes existantes.

Rapidité de calcul Toutes ces méthodes sont exploitables dans le cadre de l’édition interactive. Même dans les cas complexes où de grandes scènes comportent beaucoup de champs d’influence, les temps de calcul n’excèdent jamais la dizaine de secondes.

Contrôlabilité et prédictibilité Toutes ces méthodes sont contrôlables à l’aide de différents types de champs d’influence. Ces outils ayant un comportement facilement prédictibles, ces méthodes fournissent une grande liberté créative à l’utilisateur, qui peut facilement obtenir les résultats recherchés en peu d’essais.

Extensibilité Notre système a été pensé depuis le départ pour être facilement extensible, sans que cela implique de modifications sur les effets existants. Il est facile et rapide d’y ajouter un nouvel effet : il suffit de définir un nouveau type de champs, de créer les primitives associées, et de les lier aux objets environnementaux. Seule la seconde étape nécessite du temps de développement, les deux autres sont extrêmement rapides et intuitives.

Ces méthodes sont cependant limitées à différents niveaux. Comme elles reposent sur l’instanciation massive de modèles géométriques ou la génération massive de maillages, elles sont peu adaptées à la visualisation en temps réel. Les processus qui guident la génération des effets

ne sont pas pilotés par des règles issues du monde réel. Ces méthodes sont donc purement géométriques, ce qui implique que les résultats produits ont l'air plausibles quand ils sont observés à la distance pour laquelle ils sont conçus mais ne sont pas pour autant réalistes. De plus, notre système ne gère pas les interactions entre les effets : il est impossible de générer de la neige sur des feuilles, ou des feuilles sur la neige. Les images où l'on peut voir des feuilles mortes distribuées parmi l'herbe (Sections 4 et 5) sont produites par des primitives d'herbe modifiées afin de générer également des feuilles.

Conclusion et perspectives

Contributions

Notre modèle hiérarchique de scène composée d'objet environnemental constitue une nouvelle approche, radicalement différente, qui apporte plusieurs contributions au domaine du changement d'apparence.

Facteur d'échelle Ce système rend possible la création de scènes de plusieurs centaines de mètres carrés comportant des détails de l'ordre du centimètre, là où les méthodes existantes sont limitées à un rapport taille sur détails de l'ordre de 10^3 . Cet accroissement supérieur d'un ordre de grandeur nous permet par conséquent de produire des scènes dont la surface est environ cent fois supérieure à ce qu'il était possible de faire jusqu'à présent à détails équivalents. Contrairement aux autres méthodes qui autorisent la création soit de petites scènes très détaillées, soit de grandes scènes beaucoup moins fines, il nous permet également de mélanger des détails qui varient de 1 à 20 centimètres au sein d'une même scène.

Création de scènes complexes Les opérateurs de niveaux de détails et l'instanciation d'effets nous permettent de créer de grandes scènes riches, composées de plusieurs centaines de milliers d'objets. Le contrôle global de l'environnement permet de faire varier très facilement et très rapidement l'apparence d'une scène entière.

Contrôle Le contrôle est une contribution importante de ces travaux. Nous fournissons aux utilisateurs un moyen simple, rapide, et hautement prédictible pour créer et modifier des scènes. Les champs d'influence sont un concept simple à appréhender et à manipuler. Ce sont des outils puissants qui permettent de modéliser un vaste panel de caractéristiques tels que des traces laissées par des piétons, des animaux ou des véhicules, des tas de neige ou de feuilles, le dessèchement de l'herbe, etc.

Perspectives

Notre plateforme pourrait être étendue selon plusieurs axes.

L'étape d'enrichissement qui crée un objet environnemental à partir d'un modèle géométrique est pour l'instant entièrement manuelle et doit être réalisée par un concepteur ou un utilisateur de la plateforme. Automatiser cette étape serait une amélioration non négligeable. Une analyse de la surface du modèle géométrique d'un objet permettrait de définir quelles sont les sous-parties de l'objet affectées par un effet choisi, et permettrait d'attacher ensuite automatiquement à cet objet des primitives liées à l'effet concerné.

Notre système complètement statique ne gère aucune animation, et il serait intéressant de travailler sur cette piste. La prise en compte du vent par exemple apporterait beaucoup plus de réalisme, en permettant de simuler le transport de feuilles mortes ou en modélisant dynamiquement des congères. La combinaison d'un système de gestion de foules avec notre plateforme serait également très intéressante et permettrait de générer les détails liés à l'activité humaine (traces d'utilisation, usure, etc.).

Lors de ces travaux, nous nous sommes focalisés sur des effets qui apportent un fort changement visuel dans les scènes, et comme cela a été prévu depuis le début de la conception de cette plateforme, de nouveaux effets peuvent être y ajoutés facilement. Comme cette méthode fonctionne par objet, elle permet d'utiliser différentes techniques optimisées par objet et par phénomène. Il serait intéressant de travailler entre autre sur l'intégration du vieillissement, de la corrosion, ou de la pousse de mousse ou de lichens.

Annexe A

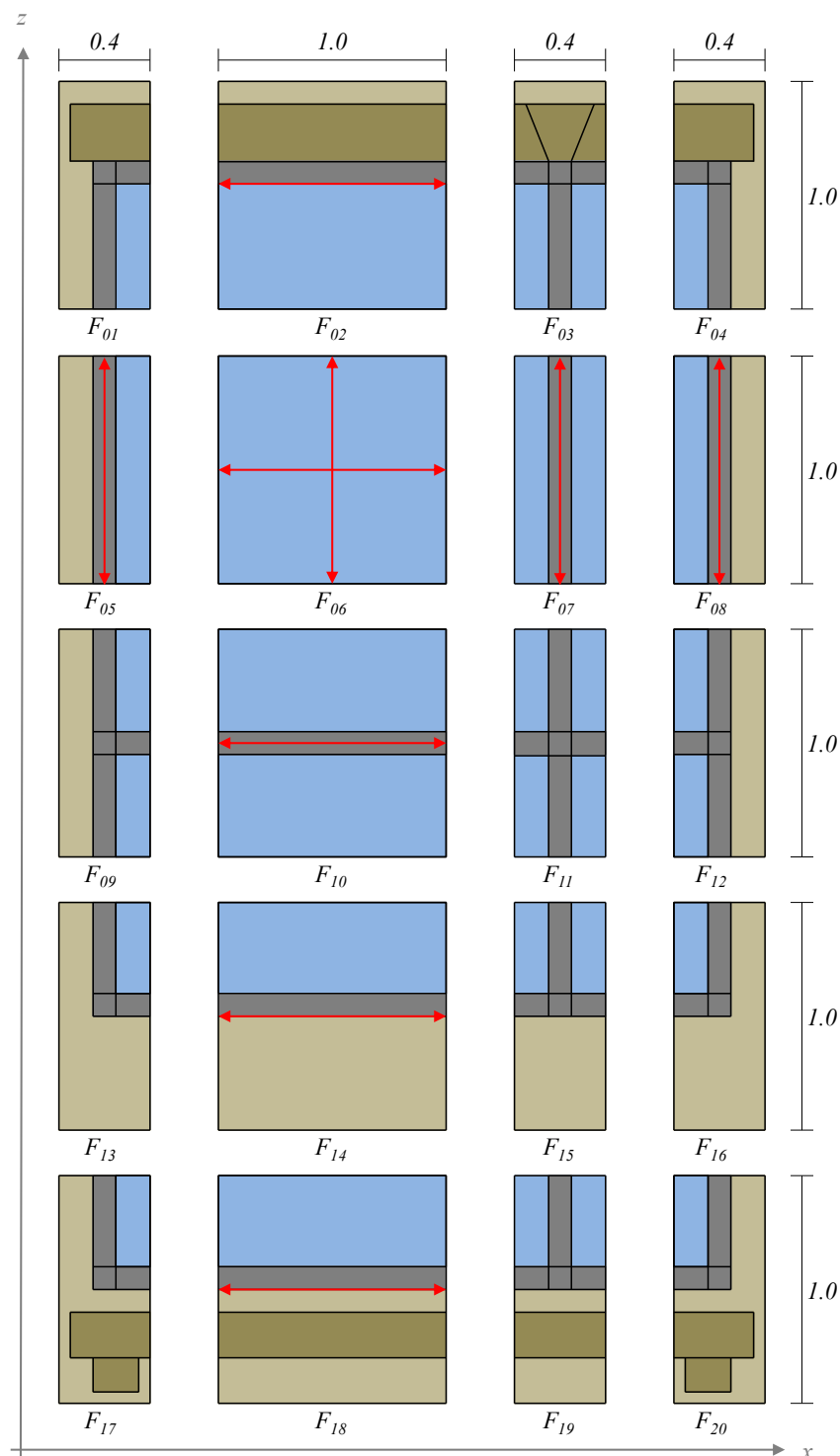
Objets environnementaux paramétrés

Rappelons que les objets environnementaux paramétrés sont des objets environnementaux qui, en plus d'être réactifs à l'environnement, sont capables de s'adapter automatiquement à des contraintes supplémentaires, par exemple de dimensions. Il y a deux types d'objets environnementaux paramétrés : soit leur géométrie est composée d'une seule composante connexe paramétrée (par exemple un parallélépipède rectangle paramétré par ses deux sommets extrêmes), soit ils sont un ensemble d'objets environnementaux, désignés dans ce contexte comme *sous-objets*. La construction d'un objet environnemental paramétré se fait en déterminant soit comment doivent être paramétrés sa géométrie et ses effets, soit comment doivent-être transformés et agencés ses différents sous-objets.



FIGURE A.1 – L'objet environnemental paramétré *fenêtre* s'adapte à différentes contraintes de dimension.

Pour décrire le fonctionnement des objets environnementaux paramétrés, analysons le cas concret d'une fenêtre (Figure A.1). Les différents sous-objets qui la composent et leurs dimensions sont listés dans la figure A.2, qui est une représentation simplifiée en deux dimensions vue de face. Les différentes couleurs des sous-objets représentent les différents matériaux. Elles sont utilisées pour illustrer comment les différents sous-objets doivent se connecter de manière cohérente.

FIGURE A.2 – Les différents sous-objets de l'objet environnemental paramétré *fenêtre*

Transformations des sous-objets

Les variations de dimensions des objets environnementaux paramétrés reposent sur le schéma d'assemblage et les variations de dimensions des sous-objets. Certains sous-objets sont conçus pour être mis à l'échelle sur un ou plusieurs axes. Pour faire varier les dimensions d'un objet environnemental paramétré, il faut donc faire varier les dimensions de certains de ses sous-objets,

et il est nécessaire que ces variations conservent la cohérence des connexions entre les sous-objets.

Les sous-objets de chacun de ces tuples (F_{13}, F_{17}) , (F_{14}, F_{18}) , (F_{15}, F_{19}) et (F_{16}, F_{20}) ont un rôle similaire : un seul élément de chaque tuple sera utilisé pour construire la fenêtre, en fonction de la présence ou de l'absence d'un balcon. Les sous-objets F_{17} , F_{18} , F_{19} et F_{20} ont un support qui sert de base au balcon, alors que les sous-objets F_{13} , F_{14} , F_{15} et F_{16} n'en ont pas. Une flèche rouge, horizontale ou verticale, indique que le sous-objet sur lequel elle est apposée est conçu pour être mis à l'échelle sur l'axe correspondant. Ainsi, F_{02} , F_{06} , F_{10} , F_{14} et F_{18} sont conçus pour être mis à l'échelle sur l'axe x , et F_{05} , F_{06} , F_{07} , et F_{08} sont conçus pour être mis à l'échelle sur l'axe z . On notera que le sous-objet F_{06} peut être mis à l'échelle sur les deux axes.

La géométrie des sous-objets est pensée pour que les mises à l'échelle autorisées sur cet objet, désignées comme *valides*, ne détruisent pas la cohérence de ses connexions (Figure A.3).

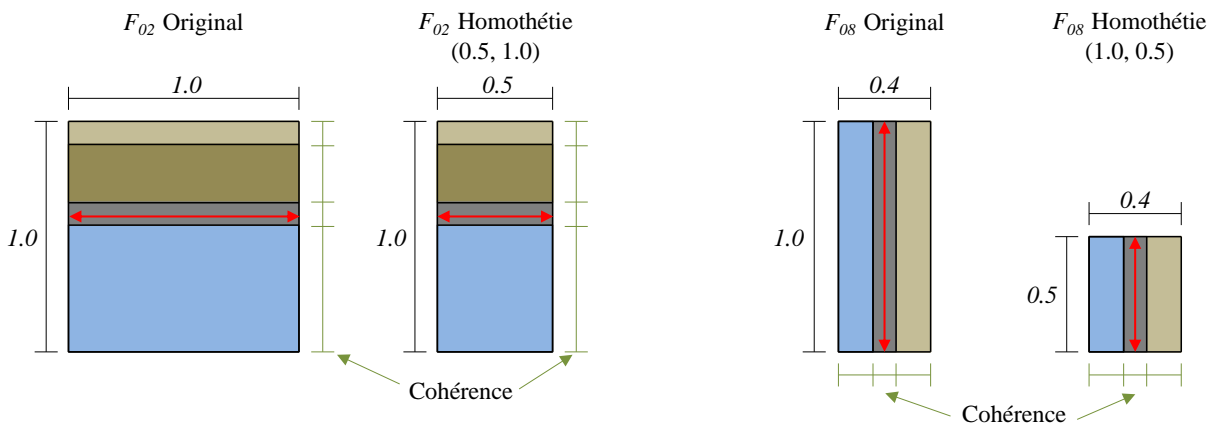


FIGURE A.3 – Mises à l'échelle valides d'un sous-objet.

À l'inverse, si on effectue une mise à l'échelle sur un sous objet qui n'a pas été conçu pour, cela empêchera l'utilisation de ce sous-objet transformé, puisqu'il ne se connectera pas correctement aux autres sous-objets (Figure A.4). Une telle transformation sera désignée comme *invalidé*.

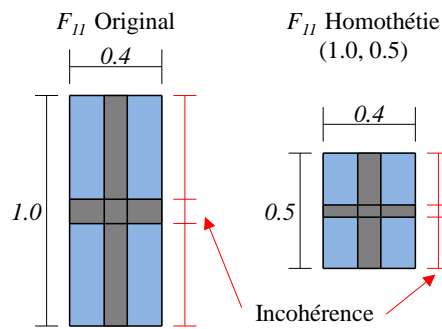


FIGURE A.4 – Mise à l'échelle invalide d'un sous-objet.

Dimensions nominales

La figure A.5 montre le schéma d'assemblage de la fenêtre dans le cas le plus simple, où tous les sous-objets sont utilisés, et où aucun ne subit de transformation. La figure Figure A.6 montre le rendu associé. La fenêtre est donc construite avec ses dimensions *nominales* : sa largeur est la

somme des largeurs de chacune des colonnes de la matrice, et il en est de même pour sa hauteur. On peut constater la réutilisation de plusieurs sous objets dans le schéma : F_{02} , F_{05} , F_{08} et F_{10} sont utilisés deux fois, et F_{06} est utilisé quatre fois. La fenêtre a pour *largeur nominale* l_{nom} 3.2 mètres et pour *hauteur nominale* h_{nom} 5.0 mètres.

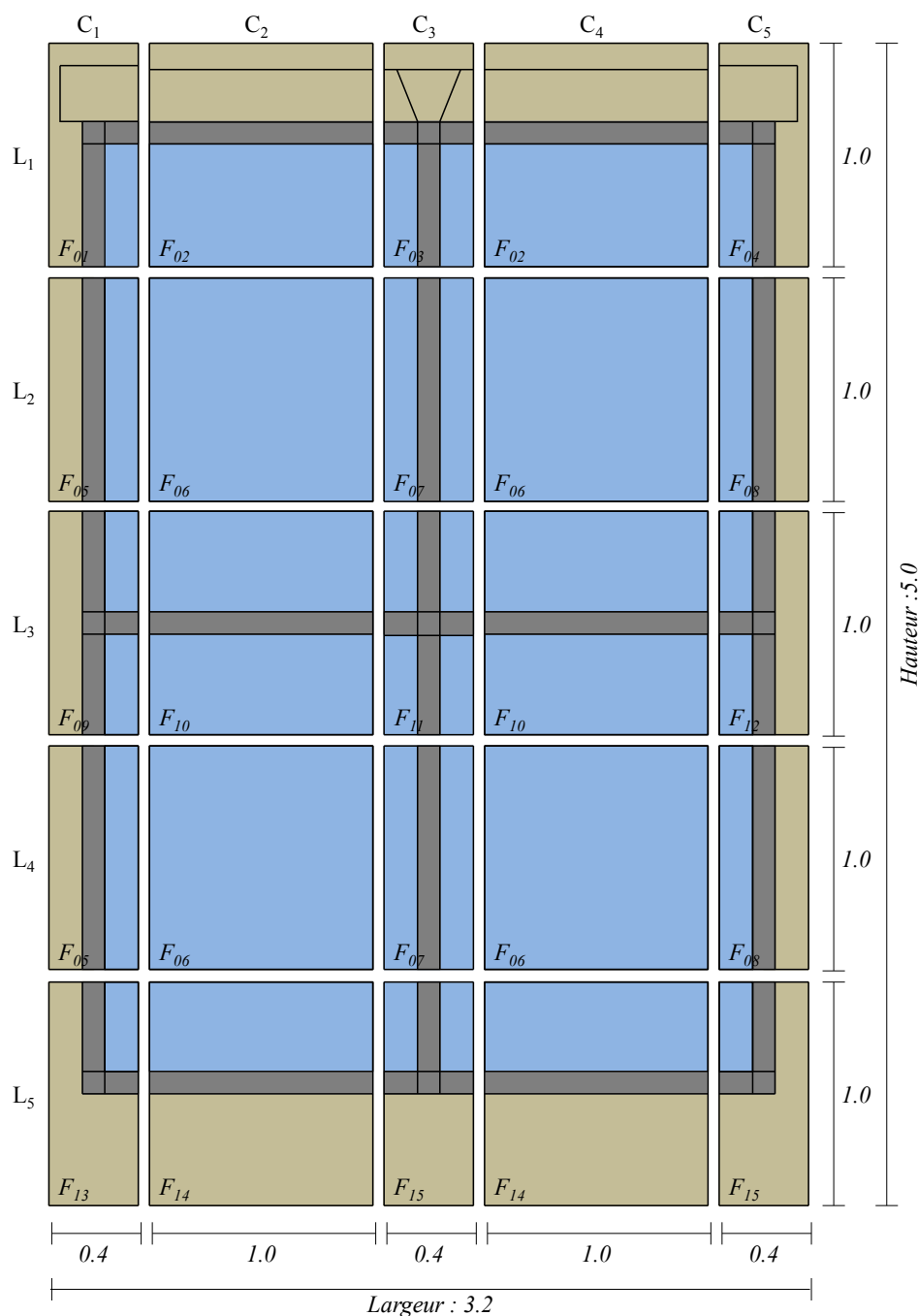


FIGURE A.5 – Schéma de la fenêtre construite avec ses dimensions nominales.

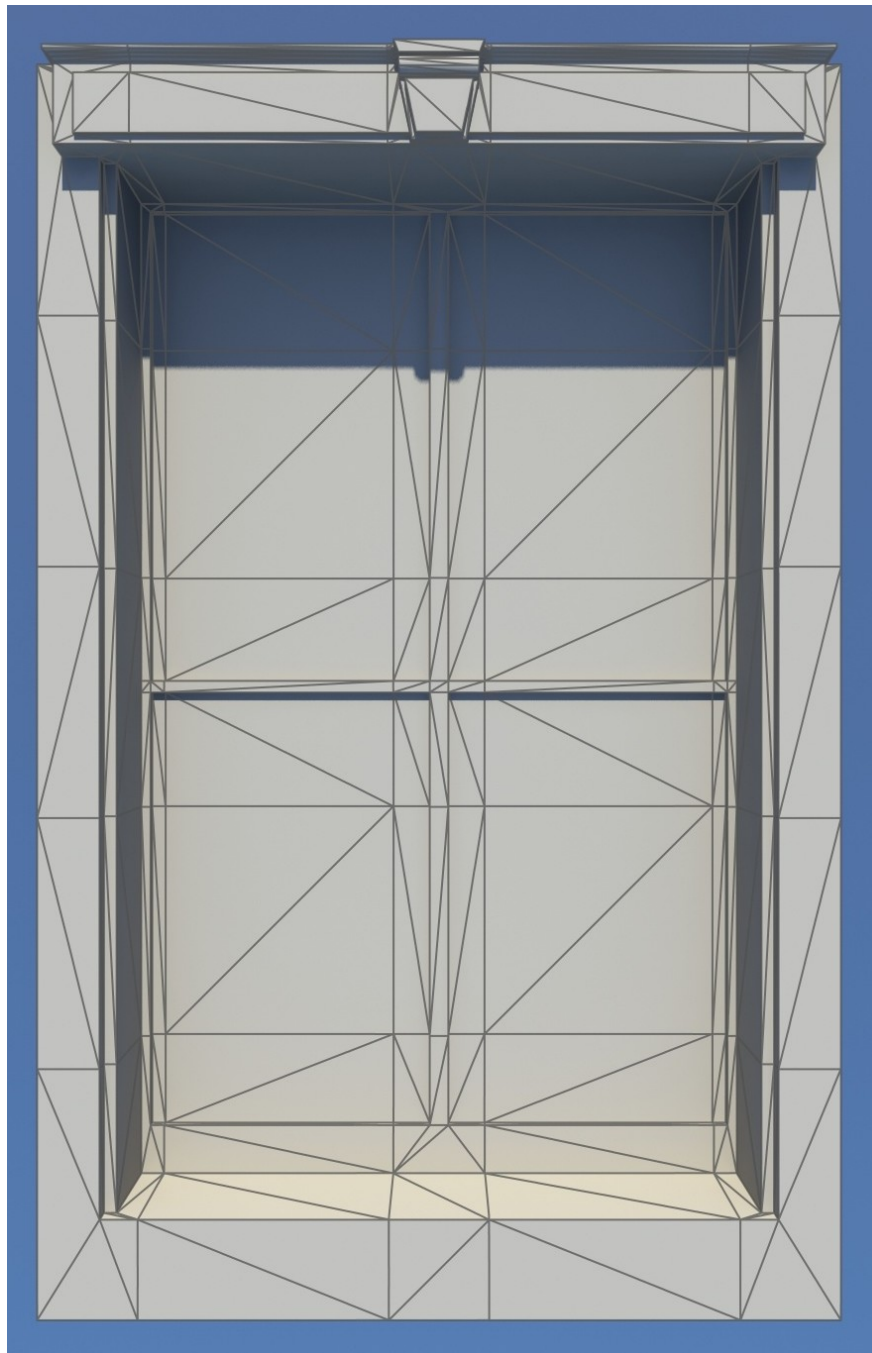


FIGURE A.6 – Rendu de la fenêtre construite avec ses dimensions nominales.

Dimensions minimales

Les dimensions *minimales* d'un objet sont le résultat de sa construction lorsque l'on applique toutes les mises à l'échelle possibles avec les paramètres minimaux. Reprenons la liste des sous-objets qui composent la fenêtre, visible dans la figure A.2. F_{02} , F_{06} , F_{10} et F_{14} vont donc être mis à l'échelle avec pour paramètre le vecteur $(0.0, 1.0)$; et F_{05} , F_{06} , F_{07} , et F_{08} vont l'être avec le vecteur $(1.0, 0.0)$. Ces mises à l'échelle correspondent à la suppression des colonnes C_2 et c_4 , puisque leur largeur est réduite à 0, ainsi que des lignes L_2 et L_4 puisque leur hauteur est également réduite à 0. Le résultat obtenu, la fenêtre ramenée à ses dimensions minimales,

est visible dans la figure Figure A.7. La fenêtre a pour *largeur minimale* l_{min} 1.2 mètres et pour *hauteur minimale* h_{min} 3.0 mètres.

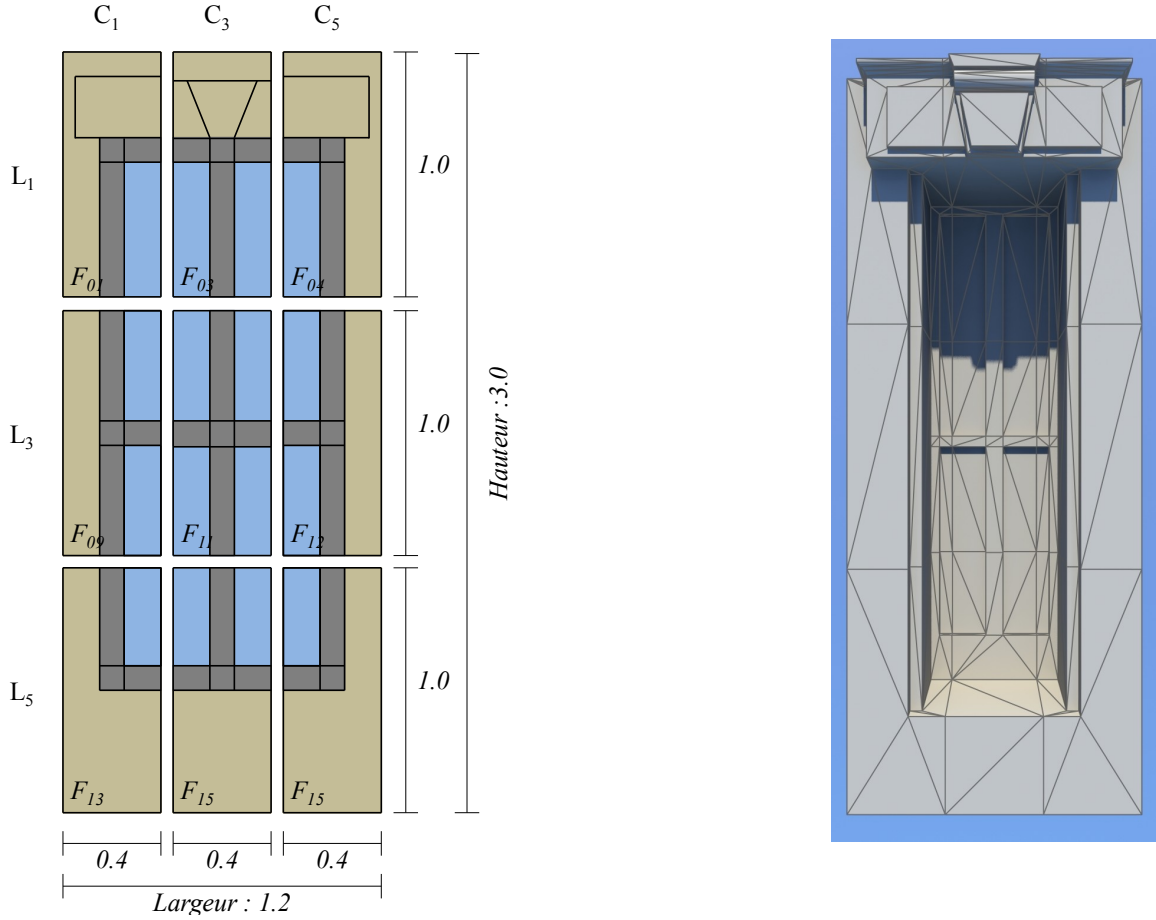


FIGURE A.7 – Fenêtre construite avec ses dimensions minimales.

Autres dimensions

La fenêtre peut être construite avec n'importe quelle largeur $l >$ largeur minimale et n'importe quelle hauteur $h >$ hauteur minimale. À partir d'une largeur l et d'une hauteur h , il est nécessaire de déterminer l_t et h_t , respectivement la largeur et la hauteur utilisées pour effectuer la transformation correspondante sur les colonnes C_2 et C_4 , ainsi que sur les lignes L_2 et L_4 . Le calcul est le suivant :

$$l_t = (l - l_{min})/2.0$$

$$h_t = (h - h_{min})/2.0$$

Il faut donc transformer les sous-objets des colonnes C_2 et C_4 avec une homothétie ayant pour paramètre le vecteur $(l_t, 1.0)$ et les sous objets des lignes L_2 et L_4 avec une homothétie ayant pour paramètre le vecteur $(1.0, h_t)$.

La fenêtre n'a pas de dimensions maximales : il est possible de la créer avec $l \gg l_{nom}$ et $h \gg h_{nom}$. L'apparence de l'objet perdra cependant en pertinence si on la compare à l'objet créé au départ, mais le modèle reste valide, et garantit que la fenêtre sera cohérente géométriquement.

Changement d'apparence

Le changement d'apparence d'un objet environnemental paramétré composé de sous-objets est transparent : étant donné que chacun des sous-objets qui le compose est un objet environnemental, chacun d'eux gère ses changements d'apparence. Le changement d'apparence d'un objet environnemental paramétré est donc l'ensemble des changements d'apparence de ses sous-objets. Il est cependant nécessaire que les conditions aux bords des différentes primitives environnementales adjacentes soient identiques, afin de ne pas créer d'incohérences lors de la génération des effets. On observe dans la partie gauche de la figure A.8 que quand les tangentes qui contrôlent les surfaces de neige sont positionnées de façon à rendre la surface de neige cohérente au niveau des sous-objets, une incohérence apparaît lors de la construction de l'objet environnemental paramétré. Au contraire, la partie droite de cette figure montre que des tangentes horizontales, qui rendent la couche de neige cohérente au niveau de l'objet environnemental paramétré, rendent la surface de neige incohérente au niveau des sous objets en eux même. Cet exemple illustre le fait que la configuration des différentes primitives environnementales d'un sous objet doit être réalisée en fonction de son rôle dans l'objet environnemental paramétré, et non à l'échelle du sous-objet seul.

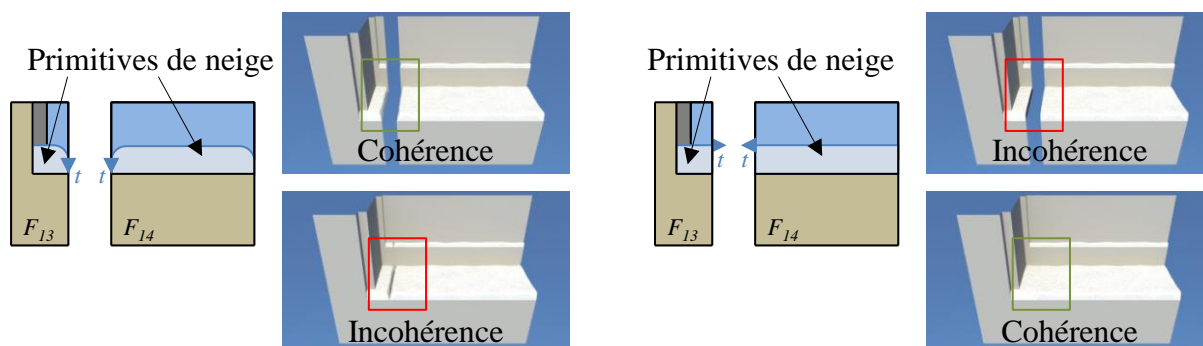


FIGURE A.8 – Influence des conditions au bord sur des primitives environnementales adjacentes.

Publications

Revue

François Grosbellet, Adrien Peytavie, Éric Guérin, Éric Galin, Stéphane Mérillou, et Bedřich Beneš

Environmental Objects for Authoring Procedural Scenes

Computer Graphics Forum – CGF

À paraître, Wiley-Blackwell

Conférences

François Grosbellet, Stéphane Mérillou, Éric Galin, Éric Guérin, Adrien Peytavie, et Bedřich Beneš

Modélisation procédurale de changements de saison

Proc. of the Journées de l'Association Française d'Informatique Graphique – AFIG

Limoges, France, 2013, IRIT Presse

François Grosbellet, Stéphane Mérillou, Éric Galin, Éric Guérin, Adrien Peytavie, et Bedřich Beneš

Modélisation procédurale de changements de saison

Second prix ex-aequo dans la catégorie poster

Workshop doctorant XLIM, Limoges, 2013

Bibliographie

- [1] Shohei ADACHI, Hiroaki UKAJI, Takahiro KOSAKA et Shigeo MORISHIMA, **Real-time dust rendering by parametric shell texture synthesis**. *Poster on ACM SIGGRAPH*, 2013. (Cité page 20)
- [2] Daniel G. ALIAGA, Carlos A. VANEGAS et Bedřich BENEŠ, **Interactive example-based urban layout synthesis**. *ACM Transactions on Graphics – TOG*, 27(5): 160, 2008. (Cité page 15)
- [3] Monssef ALSWEIS et Oliver DEUSSEN, **Modeling and visualization of symmetric and asymmetric plant competition**. *Proceedings of the First Eurographics Conference on Natural Phenomena*, pages 83 – 88. Eurographics Association, 2005. (Cité page 35)
- [4] James ARVO, David KIRK et Apollo COMPUTER, **Modeling plants with environment-sensitive automata**. *Proceedings of Ausgraph'88*, pages 27–33, 1988. (Cité page 57)
- [5] Brook BAKAY, Paul LALONDE et Wolfgang HEIDRICH, **Real-time animated grass**. *Proceedings of Eurographics*, 2002. (Cité page 38)
- [6] Aurelien BARBIER et Eric GALIN, **Fast distance computation between a point and cylinders, cones, line-swept spheres and cone-spheres**. *Journal of Graphics tools*, 9(2):11 – 19, 2004. (Cité page 59)
- [7] Welton BECKET et Norman I. BADLER, **Imperfection for realistic image synthesis**. *The Journal of Visualization and Computer Animation*, 1(1):26 – 32, 1990. (2 citations pages 18 et 22)
- [8] Bedřich BENEŠ, Ondrej ŠT'AVA, Radomír MĚCH et Gavin MILLER, **Guided procedural modeling**. *Computer Graphics Forum – CGF*, 30(2):325 – 334, 2011. (Cité page 57)
- [9] Bedřich BENEŠ, **A stable modeling of large pant ecosystems**. *Proceedings of the International Conference on Computer Vision and Graphics*, pages 94 – 101, 2002. (Cité page 35)
- [10] Bedřich BENEŠ et Javier Abdul CORDÓBA, **Modeling virtual gardens by autonomous procedural agents**. *Proceedings of Theory and Practice of Computer Graphics*, pages 58 – 65. IEEE, 2003. (Cité page 36)

- [11] Bedřich BENEŠ et Juan Miguel SOTO, **Clustering in virtual ecosystems**. *Proceedings of the 12-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 29 – 34, 2004. (Cité page [36](#))
- [12] Bedřich BENEŠ, Václav TĚŠÍNSKÝ, Jan HORNYŠ et Sanjiv K. BHATIA, **Hydraulic erosion**. *Computer Animation and Virtual Worlds – CAVW*, 17(2):99–108, 2006. (Cité page [10](#))
- [13] James F. BLINN, **Simulation of wrinkled surfaces**. *SIGGRAPH Comput. Graph.*, 12(3):286 – 292, 1978. (3 citations pages [18](#), [20](#), et [22](#))
- [14] James F. BLINN, **Light reflection functions for simulation of clouds and dusty surfaces**. *SIGGRAPH Comput. Graph.*, 16(3):21 – 29, 1982. (Cité page [20](#))
- [15] Craig F. BOHREN et Bruce R. BARKSTROM, **Theory of the optical properties of snow**. *Journal of Geophysical Research*, 79(30):4527 – 4535, 1974. (Cité page [22](#))
- [16] Laurence BOISSIEUX, Gergo KISS, Nadia Magnenat THALMANN et Prem KALRA, **Simulation of skin aging and wrinkles with cosmetics insight**. *Proceedings of the Eurographic Workshop on Computer Animation and Simulation*, pages 15 – 27, 2000. (Cité page [21](#))
- [17] Carles BOSCH, Xavier PUEYO, Stéphane MÉRILLOU et Djamchid GHAZANFARPOUR, **A physically-based model for rendering realistic scratches**. *Computer Graphics Forum – CGF*, 23(3):361 – 370, 2004. (Cité page [19](#))
- [18] Frédéric BOUDON, Alexandre MEYER et Christophe GODIN, **Survey on computer representations of trees for realistic and efficient rendering**. Rapport de recherche, LIRIS UMR 5205 CNRS / INSA de Lyon / Université Claude Bernard Lyon 1 / Université Lumière Lyon 2 / École Centrale de Lyon, 2006. (Cité page [13](#))
- [19] Kévin BOULANGER, **Real-time realistic rendering of nature scenes with dynamic lighting**. Thèse de Doctorat, Université de Rennes I, 2008. (Cité page [36](#))
- [20] Kévin BOULANGER, Sumanta N. PATTANAIK et Kadi BOUATOUCH, **Rendering grass in real time with dynamic lighting**. *IEEE Computer Graphics and Applications*, 29(1):32 – 41, 2009. (2 citations pages [7](#) et [36](#))
- [21] Michael BRAITMAIER, Joachim DIEPSTRATEN et Thomas ERTL, **Real-time rendering of seasonal influenced trees**. *Proceedings of Theory and Practice of Computer Graphics*, pages 152 – 159. IEEE, 2004. (Cité page [33](#))
- [22] Éric BRUNETON et Fabrice NEYRET, **Real-time rendering and editing of vector-based terrains**. *Computer Graphics Forum – CGF*, 27(2):311–320, 2008. (Cité page [11](#))
- [23] Lee A. BUTLER et David S. EBERT, **Procedural field grasses**. *ACM SIGGRAPH 98 Conference Abstracts and Applications*, page 270. ACM, 1998. (Cité page [37](#))

- [24] Marcio CABRAL, Sylvain LEFEBVRE, Carsten DACHSBACHER et George DRETTAKIS, **Structure-preserving reshape for textured architectural scenes**. *Computer Graphics Forum – CGF*, 28(2):469 – 480, 2009. (2 citations pages [13](#) et [14](#))
- [25] Yao-Xun CHANG et Zen-Chung SHIH, **Physically-based patination for underground objects**. *Computer Graphics Forum – CGF*, 19(3):109–117, 2000. (Cité page [21](#))
- [26] Yao-Xun CHANG et Zen-Chung SHIH, **The synthesis of rust in seawater**. *The Visual Computer*, 19(1):50–66, 2003. (Cité page [21](#))
- [27] Yanyun CHEN, Xin TONG, Jiaping WANG, Stephen LIN, Baining GUO et Heung-Yeung SHUM, **Shell texture functions**. *CM Transactions on Graphics – TOG*, 23(3):343 – 353, 2004. (Cité page [20](#))
- [28] Yanyun CHEN, Lin XIA, Tien-Tsin WONG, Xin TONG, Hujun BAO, Baining GUO et Heung-Yeung SHUM, **Visual simulation of weathering by γ -ton tracing**. *ACM Transactions on Graphics – TOG*, 24(3):1127 – 1133, 2005. (2 citations pages [18](#) et [22](#))
- [29] Zhili CHEN, Miaojun YAO, Renguo FENG et Huamin WANG, **Physics-inspired adaptive fracture refinement**. *ACM Transactions on Graphics – TOG*, 33(4): 113 :1 – 113 :7, 2014. (Cité page [19](#))
- [30] Norishige CHIBA, Shunichi OHKAWA, Kazunobu MURAOKA et Mamoru MIURA, **Visual simulation of botanical trees based on virtual heliotropism and dormancy break**. *The Journal of Visualization and Computer Animation*, 5(1):3 – 15, 1994. (Cité page [57](#))
- [31] Norishige CHIBA, Ken OHSHIDA, Kazunobu MURAOKA et Nobuji SAITO, **Visual simulation of leaf arrangement and autumn colours**. *The Journal of Visualization and Computer Animation*, 7(2):79 – 93, 1996. (Cité page [33](#))
- [32] Giliam J. P. de CARPENTIER et Rafael BIDARRA, **Interactive gpu-based procedural heightfield brushes**. *Proceedings of the International Conference on Foundations of Digital Games – FDG*, pages 55–62. ACM, 2009. (Cité page [11](#))
- [33] Philippe DECAUDIN et Fabrice NEYRET, **Volumetric billboards**. *Computer Graphics Forum – CGF*, 28(8):2079 – 2089, 2009. (Cité page [37](#))
- [34] Brett DESBENOIT, Éric GALIN et Samir AKKOUCHE, **Modeling cracks and fractures**. *The Visual Computer*, 21(8 - 10):717 – 726, 2005. (Cité page [19](#))
- [35] Brett DESBENOIT, Éric GALIN et Samir AKKOUCHE, **Simulating and modeling lichen growth**. *Computer Graphics Forum – CGF*, volume(23):341 – 350, 3. (Cité page [21](#))
- [36] Brett DESBENOIT, Éric GALIN, Samir AKKOUCHE et Jérôme GROSJEAN, **Modeling autumn sceneries**. *Proceedings of Eurographics*, pages 107 – 110. Eurographics Association, 2006. (4 citations pages [7](#), [33](#), [34](#), et [94](#))

- [37] Oliver DEUSSEN, Carsten COLDITZ, Marc STAMMINGER et George DRETTAKIS, **Interactive visualization of complex plant ecosystems**. *Proceedings of the conference on Visualization*, pages 219 – 226. IEEE Computer Society, 2002. (Cité page [36](#))
- [38] Oliver DEUSSEN, Pat HANRAHAN, Bernd LINTERMANN, Radomír MĚCH, Matt PHARR et Przemyslaw PRUSINKIEWICZ, **Realistic modeling and rendering of plant ecosystems**. *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 275 – 286. ACM, 1998. (4 citations pages [34](#), [35](#), [57](#), et [86](#))
- [39] Oliver DEUSSEN et Bernd LINTERMANN, **Digital design of nature : computer generated plants and organics**. Springer Science & Business Media, 2006. (Cité page [13](#))
- [40] Julie DORSEY, Alan EDELMAN, Henrik Wann JENSEN, Justin LEGAKIS et Hans Køhling PEDERSEN, **Modeling and rendering of weathered stone**. *Proceedings of SIGGRAPH Conference*, pages 225–234. ACM, 1999. (Cité page [21](#))
- [41] Julie DORSEY et Pat HANRAHAN, **Modeling and rendering of metallic patinas**. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 387 – 396. ACM, 1996. (2 citations pages [19](#) et [21](#))
- [42] Julie DORSEY, Holly E RUSHMEIER et François SILLION, **Digital Modeling of Material Appearance**. Morgan Kaufmann, 2007. (Cité page [18](#))
- [43] David S. EBERT, Forest Kenton MUSGRAVE, Darwyn PEACHEY, Ken PERLIN et Steven WORLEY, **Texturing and Modeling : A Procedural Approach**. The Morgan Kaufmann Series in Computer Graphics. Elsevier, 3rd édition, 1998. (Cité page [10](#))
- [44] Gregory ESCH, Peter WONKA, Pascal MÜLLER et Eugene ZHANG, **Interactive procedural street modeling**. *Transaction on Graphics – TOG*, 27(3):103 :1–103 :10, 2008. (Cité page [15](#))
- [45] Paul FEARING, **Computer modelling of fallen snow**. *Proceedings of SIGGRAPH Conference*, pages 37–46. ACM, 2000. (2 citations pages [23](#) et [24](#))
- [46] Bryan E. FELDMAN et James F. O'BRIEN, **Modeling the accumulation of wind-driven snow**. *ACM SIGGRAPH 2002 Conference abstracts and applications – SIGGRAPH '02*, pages 218 – 218. ACM, 2002. (Cité page [24](#))
- [47] David FOLDES et Bedřich BENEŠ, **Occlusion-based snow accumulation simulation**. *Proceedings of Virtual Reality Interactions and Physical Simulations – VRIPHYS*, pages 35–41. Eurographics Association, 2007. (2 citations pages [25](#) et [28](#))
- [48] Jonathan GAGNON et Eric PAQUETTE, **Procedural and interactive icicle modeling**. *The Visual Computer*, 27(6 – 8):451 – 461, 2011. (3 citations pages [7](#), [31](#), et [32](#))

- [49] James E. GAIN, Patrick MARAIS et Wolfgang STRASSER, **Terrain sketching**. *Proceedings of the Symposium on Interactive 3D Graphics and Games – I3D*, pages 31–38. ACM, 2009. (Cité page [11](#))
- [50] Éric GALIN, Adrien PEYTAVIE, Éric GUÉRIN et Bedřich BENEŠ, **Authoring hierarchical road networks**. *Computer Graphics Forum – CGF*, 30(7):2021–2030, 2011. (Cité page [14](#))
- [51] Éric GALIN, Adrien PEYTAVIE, Éric GUÉRIN et Nicolas MARÉCHAL, **Procedural generation of roads**. *Computer Graphics Forum – CGF*, 29(2):429–438, 2010. (Cité page [14](#))
- [52] Jean-David GÉNEVAUX, **Représentation, modélisation et génération procédurale de terrains**. Thèse de Doctorat, Université Lumière Lyon 2, 2015. (Cité page [10](#))
- [53] Jean-David GÉNEVAUX, Éric GALIN, Éric GUÉRIN, Adrien PEYTAVIE et Bedřich BENEŠ, **Terrain generation using procedural models based on hydrology**. *Transaction on Graphics – TOG*, 32(4):143 :1–143 :13, 2013. (Cité page [12](#))
- [54] Jean-David GÉNEVAUX, Éric GALIN, Adrien PEYTAVIE, Éric GUÉRIN, Cyril BRIQUET, François GROSBELLET et Bedřich BENEŠ, **Terrain modeling from feature primitives**. *Computer Graphics Forum – CGF*, 2015. (Cité page [12](#))
- [55] Loeïz GLONDU, Lien MUGUERCIA, Maud MARCHAL, Carles BOSCH, Holly RUSHMEIER, Georges DUMONT et George DRETTAKIS, **Example-based fractured appearance**. *Computer Graphics Forum – CGF*, 31(4):1547 – 1556, 2012. (Cité page [19](#))
- [56] N. GREENE, **Voxel space automata : Modeling with stochastic growth processes in voxel space**. *SIGGRAPH Comput. Graph.*, 23(3):175 – 184, 1989. (Cité page [57](#))
- [57] Sylvain GUERRAZ, Frank PERBET, David RAULO, François FAURE et Marie-Paule CANI, **A procedural approach to animate interactive natural sceneries**. *Proceedings of the 16th International Conference on Computer Animation and Social Agents*, pages 73 – 78. IEEE, 2003. (Cité page [38](#))
- [58] Ralf HABEL, Michael WIMMER et Stefan JESCHKE, **Instant animated grass**. *Journal of WSCG*, 15(1-3):123 – 128, 2007. (Cité page [38](#))
- [59] James Scott HANAN, **Parametric L-systems and Their Application to the Modelling and Visualization of Plants**. Thèse de Doctorat, The University of Regina (Canada), 1992. (Cité page [57](#))
- [60] Tommy HINKS et Ken MUSETH, **Wind-driven snow buildup using a level set approach**. *Proceedings of Eurographics Ireland Workshop Series*, pages 19–26. ACM, 2009. (4 citations pages [7](#), [24](#), [28](#), et [32](#))

- [61] Koichi HIROTA, Yasuyuki TANOUE et Toyohisa KANEKO, **Generation of crack patterns with a physical model**. *The Visual Computer*, 14(3):126 – 137, 1998. (Cité page [19](#))
- [62] Houssam HNAIDI, Éric GUÉRIN, Samir AKKOUCHE, Adrien PEYTAVIE et Éric GALIN, **Feature based terrain generation using diffusion equation**. *Computer Graphics Forum – CGF*, 29(7):2179–2186, 2010. (Cité page [11](#))
- [63] Siu-chi HSU et Tien-tsin WONG, **Simulating dust accumulation**. *IEEE Comput. Graph. Appl.*, 15(1):18 – 22, 1995. (Cité page [20](#))
- [64] Hayley N. IBEN et James F. O'BRIEN, **Generating surface crack patterns**. *Graphical Models*, 71(6):198 – 208, 2009. (Cité page [19](#))
- [65] Klemens JAHRMANN et Michael WIMMER, **Interactive grass rendering using real-time tessellation**. *WSCG*, pages 114 – 122, 2013. (Cité page [37](#))
- [66] SoHyeon JEONG, Si-Hyung PARK et Chang-Hun KIM, **Simulation of morphology changes in drying leaves**. *Computer Graphics Forum – CGF*, 32(1):204 – 215, 2013. (2 citations pages [33](#) et [94](#))
- [67] Alex D. KELLEY, Michael C. MALIN et Gregory M. NIELSON, **Terrain simulation using a model of stream erosion**. *Computer Graphics*, 22(4):263–268, 1988. (Cité page [12](#))
- [68] Dany KHARITONSKY et Jakob GONCZAROWSKI, **A physically based model for icicle growth**. *The Visual Computer*, 10(2):88 – 100, 1993. (Cité page [31](#))
- [69] Theodore KIM, David ADALSTEINSSON et Ming C LIN, **Modeling ice dynamics as a thin-film stefan problem**. *Proceedings of the 2006 ACM SIGGRAPH / Eurographics symposium on Computer animation*, pages 167 – 176. Eurographics Association, 2006. (Cité page [31](#))
- [70] Theodore KIM, Michael HENSON et Ming C LIN, **A hybrid algorithm for modeling ice formation**. *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation – SCA '04*, pages 305 – 314. Eurographics Association, 2004. (Cité page [30](#))
- [71] Theodore KIM et Ming C LIN, **Visual simulation of ice crystal growth**. *Proceedings of the 2003 ACM SIGGRAPH / Eurographics symposium on Computer animation*, pages 86 – 97. Eurographics Association, 2003. (Cité page [30](#))
- [72] Brendan LANE, Przemyslaw PRUSINKIEWICZ *et al.*, **Generating spatial distributions for multilevel models of plant communities**. *Proceedings of Graphics Interface*, pages 69 – 80. Citeseer, 2002. (Cité page [36](#))
- [73] Michael S. LANGER, Linqiao ZHANG, Allison W. KLEIN, Aditya BHATIA, Javeen PEREIRA et Dipinder REKHI, **A spectral-particle hybrid method for rendering falling snow**. *Proceedings of the Eurographics Symposium on Rendering – EGSR*, volume 4, pages 217–226. Eurographics Association, 2004. (Cité page [28](#))

- [74] Aristid LINDENMAYER, **Mathematical models for cellular interactions in development i. filaments with one-sided inputs.** *Journal of Theoretical Biology*, 18(3):280 – 299, 1968. (Cité page [12](#))
- [75] Aristid LINDENMAYER, **Mathematical models for cellular interactions in development ii. simple and branching filaments with two-sided inputs.** *Journal of Theoretical Biology*, 18(3):300 – 315, 1968. (Cité page [12](#))
- [76] William E. LORENSEN et Harvey E. CLINE, **Marching cubes : A high resolution 3d surface construction algorithm.** *ACM SIGGRAPH Computer Graphics*, volume 21, pages 163–169, 1987. (Cité page [83](#))
- [77] Nicolas MARÉCHAL, Éric GUÉRIN, Éric GALIN, Stéphane MÉRILLOU et Nicolas MÉRILLOU, **Heat transfer simulation for modeling realistic winter sceneries.** *Computer Graphics Forum – CGF*, 29(2):449–458, 2010. (5 citations pages [26](#), [27](#), [32](#), [39](#), et [81](#))
- [78] Aurélien MARTINET, Éric GALIN, Brett DESBENOIT et Samir AKKOUCE, **Procedural modeling of cracks and fractures.** *Proceedings of Shape Modeling Applications*, pages 346 – 349. IEEE, 2004. (Cité page [19](#))
- [79] James MCCRAE et Karan SINGH, **Sketch-based path design.** *Proceedings of Graphics Interface 2009*, pages 95 – 109. Canadian Information Processing Society, 2009. (2 citations pages [13](#) et [14](#))
- [80] Radomír MECH et Przemyslaw PRUSINKIEWICZ, **Visual models of plants interacting with their environment.** *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 397 – 410. ACM, 1996. (2 citations pages [13](#) et [57](#))
- [81] Àlex MÉNDEZ-FELIU et Mateu SBERT, **From obscurances to ambient occlusion : A survey.** *The Visual Computer*, 25(2):181 – 196, 2009. (Cité page [63](#))
- [82] Nicolas MÉRILLOU, Stéphane MÉRILLOU, Éric GALIN et Djamchid GHAZANFARPOUR, **Simulating how salt decay ages buildings.** *IEEE computer graphics and applications*, 32(2):44 – 54, 2011. (2 citations pages [21](#) et [57](#))
- [83] Nicolas MÉRILLOU, Stéphane MÉRILLOU, Djamchid GHAZANFARPOUR, Jean-Michel DISCHLER et Éric GALIN, **Simulating atmospheric pollution weathering on buildings.** *Proceedings of the International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision – WSCG*, pages 65 – 72. Václav Skala-UNION Agency, 2010. (Cité page [20](#))
- [84] Stéphane MÉRILLOU, Jean-Michel DISCHLER et Djamchid GHAZANFARPOUR, **Corrosion : simulating and rendering.** *Proceedings of Graphics Interface*, pages 167 – 174, 2001. (Cité page [21](#))
- [85] Stéphane MÉRILLOU, Jean-Michel DISCHLER et Djamchid GHAZANFARPOUR, **Surface scratches : measuring, modeling and rendering.** *The Visual Computer*, 17(1):30 – 45, 2001. (Cité page [19](#))

- [86] Paul MERRELL, Eric SCHKUFZA et Vladlen KOLTUN, **Computer-generated residential building layouts**. *ACM Transactions on Graphics – TOG*, 29(6):181–191, 2010. (Cité page [17](#))
- [87] Elie MICHEL, Arnaud EMILIEN et Marie-Paule CANI, **Generation of folded terrains from simple vector maps**. *Short at Eurographics – EG*. Eurographics Association, 2015. (2 citations pages [10](#) et [11](#))
- [88] Gavin MILLER, **Efficient algorithms for local and global accessibility shading**. *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 319 – 326. ACM, 1994. (Cité page [63](#))
- [89] Stéphane MÉRILLOU et Djamchid GHAZANFARPOUR, **A survey of aging and weathering phenomena in computer graphics**. *Computers & Graphics – C&G*, 32(2):159 – 174, 2008. (Cité page [18](#))
- [90] Lien MUGUERCIA, Carles BOSCH et Gustavo PATOW, **Fracture modeling in computer graphics**. *Computers & Graphics*, 45:86 – 100, 2014. (Cité page [19](#))
- [91] Pascal MÜLLER, Peter WONKA, Simon HAEGLER, Andreas ULMER et Luc VAN GOOL, **Procedural modeling of buildings**. *Transaction on Graphics – TOG*, 25(3):614–623, 2006. (2 citations pages [16](#) et [57](#))
- [92] Pascal MÜLLER, Gang ZENG, Peter WONKA et Luc VAN GOOL, **Image-based procedural modeling of facades**. *ACM Transactions on Graphics – TOG*, 26(3): 85 – 93, 2007. (Cité page [17](#))
- [93] Lars MUNDERMANN, Peter MACMURCHY, Juraj PIVOVAROV et Przemyslaw PRUSINKIEWICZ, **Modeling lobed leaves**. *Proceedings of Computer Graphics International*, pages 60 – 65. IEEE, 2003. (2 citations pages [32](#) et [94](#))
- [94] Kazunobu MURAOKA et Norishige CHIBA, **Visual simulation of snowfall, snow cover and snowmelt**. *Proc of the International Conference on Parallel and Distributed Systems – ICPADS*, pages 187–194. IEEE, 2000. (Cité page [23](#))
- [95] Forest Kenton MUSGRAVE, Craig E. KOLB et Robert S. MACE, **The synthesis and rendering of eroded fractal terrains**. *Computer Graphics*, 23(3):41–50, 1989. (Cité page [12](#))
- [96] Mattia NATALI, Endre M. LIDAL, Julius PARULEK, Ivan VIOLA et Daniel PATEL, **Modeling terrains and subsurface geology**. *Proceedings of Eurographics State Of The Art Report – EG STAR*, pages 155–173. Eurographics Association, 2013. (Cité page [10](#))
- [97] Tomoyuki NISHITA, Hiroshi IWASAKI, Yoshinori DOBASHI et Eihachiro NAKAMAE, **A modeling and rendering method for snow by using metaballs**. *Computer Graphics Forum – CGF*, 16(3):357–364, 1997. (Cité page [23](#))
- [98] James F O’BRIEN, Adam W BARGTEIL et Jessica K HODGINS, **Graphical modeling and animation of ductile fracture**. *ACM Transactions on Graphics – TOG*, 21(3):291 – 294, 2002. (Cité page [19](#))

- [99] James F O'BRIEN et Jessica K HODGINS, **Graphical modeling and animation of brittle fracture**. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 137 – 146. ACM, 1999. (Cité page [19](#))
- [100] Per OHLSSON et Stefan SEIPEL, **Real-time rendering of accumulated snow**. *Sigrad Conference*, pages 25 – 32, 2004. (Cité page [26](#))
- [101] Jens ORTHMANN, Christof REZK-SALAMA et Andreas KOLB, **Gpu-based responsive grass**. *Journal of WSCG*, 17:65 – 72, 2009. (Cité page [37](#))
- [102] Wojciech PALUBICKI, Kipp HOREL, Steven LONGAY, Adam RUNIONS, Brendan LANE, Radomír MĚCH et Przemyslaw PRUSINKIEWICZ, **Self-organizing tree models for image synthesis**. *ACM Transactions on Graphics – TOG*, 28(3):58 – 67, 2009. (Cité page [13](#))
- [103] Eric PAQUETTE, Pierre POULIN et George DRETTAKIS, **Surface aging by impacts**. *Proceedings of Graphics Interface*, pages 175 – 182. Canadian Information Processing Society, 2001. (Cité page [19](#))
- [104] Yoav I. H. PARISH et Pascal MÜLLER, **Procedural modeling of cities**. *Proceedings of SIGGRAPH Conference – SIGGRAPH*, pages 301–308. ACM, 2001. (3 citations pages [15](#), [16](#), et [17](#))
- [105] Kurt PELZER, **Rendering countless blades of waving grass**. *GPU Gems*, 1:107 – 121, 2004. (Cité page [37](#))
- [106] Frank PERBET et Marie-Paule CANI, **Animating prairies in real-time**. *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 103 – 110. ACM, 2001. (Cité page [38](#))
- [107] Alexandre PEYRAT, Olivier TERRAZ, Séphane MÉRILLOU et Éric GALIN, **Generating vast varieties of realistic leaves with parametric 2gmap l-systems**. *The visual Computer*, 24(7 - 9):807 – 816, 2008. (2 citations pages [32](#) et [94](#))
- [108] Adrien PEYTAVIE, Éric GALIN, Stephane MÉRILLOU et Jérôme GROSJEAN, **Arches : a framework for modeling complex terrains**. *Computer Graphics Forum – CGF*, 28(2):457–467, 2009. (2 citations pages [11](#) et [95](#))
- [109] Adrien PEYTAVIE, Éric GALIN, Stephane MÉRILLOU et Jérôme GROSJEAN, **Procedural generation of rock piles using aperiodic tiling**. *Computer Graphics Forum – CGF*, 28(7):1801–1810, 2009. (Cité page [96](#))
- [110] Bui Tuong PHONG, **Illumination for computer generated pictures**. *Commun. ACM*, 18(6):311 – 317, 1975. (Cité page [20](#))
- [111] Sören PIRK, Ondrej STAVA, Julian KRATT, Michel Abdul Massih SAID, Boris NEUBERT, Radomír MĚCH, Bedřich BENEŠ et Oliver DEUSSEN, **Plastic trees : interactive self-adapting botanical tree models**. *ACM Transactions on Graphics – TOG*, 31(4):1 – 10, 2012. (2 citations pages [13](#) et [57](#))

- [112] Simon PREMOŽE, William B. THOMPSON et Peter SHIRLEY, **Geospecific rendering of alpine terrain**. *Proceedings of the Workshop on Rendering Techniques – EGWR*. Eurographics Association, 1999. (Cité page [25](#))
- [113] Przemyslaw PRUSINKIEWICZ, **Graphical applications of l-systems**. *Proceedings on Graphics Interface '86 / Vision Interface '86*, pages 247 – 253. Canadian Information Processing Society, 1986. (Cité page [12](#))
- [114] Przemyslaw PRUSINKIEWICZ et Marc HAMMEL, **A fractal model of mountains with rivers**. *Proceedings of Graphics Interface – GI*, pages 174–180. Canadian Information Processing Society, 1993. (Cité page [12](#))
- [115] Przemyslaw PRUSINKIEWICZ, Mark JAMES et Radomír MĚCH, **Synthetic topiary**. *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 351 – 358, 1994. (2 citations pages [13](#) et [57](#))
- [116] Hang QIU, Leiting CHEN, Jim X. CHEN et Yugang LIU, **Dynamic simulation of grass field swaying in wind**. *Journal of Software*, 7(2):431 – 439, 2012. (Cité page [38](#))
- [117] Hang QIU, Leiting CHEN et Guo-Ping QIU, **A novel approach to simulate the interaction between grass and dynamic objects**. *WSEAS Transactions on Computers*, 12(7):277 – 287, 2013. (Cité page [38](#))
- [118] Pascale ROUDIER, Bernard PEROCHE et Michel PERRIN, **Landscapes synthesis achieved through erosion and deposition process simulation**. *Computer Graphics Forum – CGF*, 12(3):375–383, 1993. (Cité page [10](#))
- [119] Masaki SATO, Jun KOBAYASHI, Tomoaki MORIYA, Yuki MORIMOTO et Tokiichiro TAKAHASHI, **An icicle generation model based on the sph method**. *Poster on ACM SIGGRAPH*, 2014. (Cité page [32](#))
- [120] Henry SCHÄFER, Benjamin KEINERT, Matthias NIESSNER, Christoph BUCHENAU, Michael GUTHE et Marc STAMMINGER, **Real-time deformation of subdivision surfaces from object collisions**. *Proceedings of the 6th High-Performance Graphics Conference*, pages 1 – 8. Eurographics association, 2014. (Cité page [28](#))
- [121] Musawir A SHAH, Jaakko KONTINEN et Sumanta PATTANAIK, **Real-time rendering of realistic-looking grass**. *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 77 – 82. ACM, 2005. (Cité page [37](#))
- [122] Peter SHIRLEY, Michael ASHIKHMIN et Steve MARSCHNER, **Fundamentals of computer graphics**. CRC Press, 2009. (Cité page [59](#))
- [123] Ruben M. SMELIK, Tim TUTENEL, Rafael BIDARRA et Bedřich BENEŠ, **A survey on procedural modelling for virtual worlds**. *Computer Graphics Forum – CGF*, 33(6):31–50, 2014. (Cité page [10](#))
- [124] G STINY, **Introduction to shape and shape grammars**. *Environment and Planning B : Planning and Design*, 7(3):343 – 351, 1980. (Cité page [16](#))

- [125] Alexey STOMAKHIN, Craig SCHROEDER, Lawrence CHAI, Joseph TERAN et Andrew SELLE, **A material point method for snow simulation**. *ACM Transactions on Graphics – TOG*, 32(4):102 :1 – 102 :10, 2013. (Cité page [28](#))
- [126] Robert SUMNER, **Pattern formation in lichen**. Thèse de Doctorat, Massachusetts Institute of Technology, 2001. (Cité page [21](#))
- [127] Robert W. SUMNER, James F. O'BRIEN et Jessica K. HODGINS, **Animating sand, mud and snow**. *Proceedings of Graphics Interface – GI*, pages 125–132. Canadian Information Processing Society, 1998. (Cité page [28](#))
- [128] K. SZILDER et E.P. LOZOWSKI, **Simulation of icicle growth using a three-dimensional random walk model**. *Atmospheric Research*, 36(3-4):243 – 249, 1995. (Cité page [31](#))
- [129] Jerry O TALTON, Yu LOU, Steve LESSER, Jared DUKE, Radomír MĚCH et Vladlen KOLTUN, **Metropolis procedural modeling**. *ACM Transactions on Graphics – TOG*, 30(2):1 – 14, 2011. (Cité page [13](#))
- [130] Flora Ponjou TASSE, Arnaud EMILIEN, Marie-Paule CANI, Stefanie HAHMANN et Neil DODGSON, **Feature-based terrain editing from complex sketches**. *Computers & Graphics – C&G*, 45:101–115, 2014. (Cité page [11](#))
- [131] Soon Tee TEOH, **RiverLand : An efficient procedural modeling system for creating realistic-looking terrains**. *Proceedings of the International Symposium on Advances in Visual Computing – ISVC*, pages 468–479. Springer, 2009. (Cité page [12](#))
- [132] Yusuke TSUDA, Yonghao YUE, Yoshinori DOBASHI et Tomoyuki NISHITA, **Visual simulation of mixed-motion avalanches with interactions between snow layers**. *The Visual Computer*, volume(26):883 – 891, 6 – 8. (Cité page [22](#))
- [133] Carlos A. VANEGAS, Daniel G. ALIAGA, Bedřich BENEŠ et Paul A. WADDELL, **Interactive design of urban spaces using geometrical and behavioral modeling**. *ACM Transactions on Graphics – TOG*, 28(5):111–120, 2009. (Cité page [15](#))
- [134] Niels von FESTENBERG et Stefan GUMHOLD, **A geometric algorithm for snow distribution in virtual scenes**. *Proceedings of the Eurographics Workshop on Natural Phenomena – EGWNP*, pages 15–22. Eurographics Association, 2009. (Cité page [27](#))
- [135] Niels von FESTENBERG et Stefan GUMHOLD, **Diffusion-based snow cover generation**. *Computer Graphics Forum – CGF*, 30(6):1837–1849, 2011. (Cité page [27](#))
- [136] Ondřej ŠT'AVA, Bedřich BENEŠ, Matthew BRISBIN et Jaroslav KŘIVÁNEK, **Interactive terrain modeling using hydraulic erosion**. *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 201–210. Eurographics Association, 2008. (Cité page [10](#))
- [137] Changbo WANG, Zhangye WANG, Tian XIA et Qunsheng PENG, **Real-time snowing simulation**. *The Visual Computer*, 22(5):315–323, 2006. (Cité page [24](#))

- [138] Stephen G. WARREN et Warren J. WISCOMBE, **A model for the spectral albedo of snow. ii : Snow containing atmospheric aerosols.** *Journal of the Atmospheric Sciences*, 37(12):2734–2745, 1980. (Cité page [22](#))
- [139] Basil WEBER, Pascal MÜLLER, Peter WONKA et Markus GROSS, **Interactive geometric simulation of 4d cities.** *Computer Graphics Forum – CGF*, 28(2):481 – 492, 2009. (Cité page [15](#))
- [140] Xiaoming WEI, Ye ZHAO, Zhe FAN, Wei LI, Suzanne YOAKUM-STOVER et Arie KAUFMAN, **Blowing in the wind.** *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 75 – 85. Eurographics Association, 2003. (Cité page [34](#))
- [141] Jakub WEJCHERT et David HAUMANN, **Animation aerodynamics.** *SIGGRAPH Comput. Graph.*, 25(4):19 – 22, 1991. (Cité page [34](#))
- [142] Warren J. WISCOMBE et Stephen G. WARREN, **A model for the spectral albedo of snow. i : Pure snow.** *Journal of the Atmospheric Sciences*, 37(12):2712 – 2733, 1980. (Cité page [22](#))
- [143] Tien-Tsin WONG, Wai-Yin NG et Pheng-Ann HENG, **A geometry dependent texture generation framework for simulating surface imperfections.** *Proceedings of the Eurographics Workshop on Rendering Techniques '97*, pages 139 – 150. Springer-Verlag, 1997. (Cité page [20](#))
- [144] Peter WONKA, Michael WIMMER, François X. SILLION et William RIBARSKY, **Instant architecture.** *ACM Transactions on Graphics*, 22(4):669 – 677, 2003. (2 citations pages [16](#) et [17](#))
- [145] Yin WU, Prem KALRA, Laurent MOCCOZET et Nadia MAGNENAT-THALMANN, **Simulating wrinkles and skin aging.** *The visual computer*, 15(4):183 – 198, 1999. (Cité page [20](#))
- [146] Brian WYVILL, Andrew GUY et Éric GALIN, **Extending the csg tree. warping, blending and boolean operations in an implicit surface modeling system.** *Computer Graphics Forum – CGF*, 18(2):149 – 158, 1999. (4 citations pages [58](#), [59](#), [63](#), et [83](#))
- [147] Chen YANYUN, Hanqiu SUN, Lin HUI et Enhua WU, **Modelling and rendering of snowy natural scenery using multi-mapping techniques.** *The Journal of Visualization and Computer Animation*, 14(1):21 – 30, 2003. (Cité page [28](#))
- [148] Xiangkun ZHAO, Fengxia LI et Shouyi ZHAN, **Real-time animating and rendering of large scale grass scenery on gpu.** *Proceedings of the International Conference on Information Technology and Computer Science – ITCS 2009*, pages 601 – 604. IEEE, 2009. (Cité page [37](#))
- [149] Ning ZHOU, Weiming DONG et Xing MEI, **Realistic simulation of seasonal variant maples.** *Proceedings of the Second International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications – PMA*, pages 295 – 301. IEEE, 2006. (Cité page [33](#))

Génération de détails dans les mondes procéduraux

La génération de mondes virtuels est un domaine de recherche très actif en informatique graphique : la modélisation de plantes, d'arbres, de bâtiments, de villes ou de terrains, et les simulations de vieillissement sont des domaines très explorés. Dans ce contexte, les changements d'apparences constituent également un domaine de recherche important, de part leur impact majeur dans le réalisme des scènes virtuelles produites. Ces travaux se concentrent sur la mise au point d'approches procédurales permettant de représenter les changements d'apparence sous la forme de décorations géométriques (accumulation de neige ou formation de glace, dépôt de feuilles mortes, etc.) à la fois à grande échelle et avec un très haut niveau de détail.

Nous proposons d'abord un modèle d'organisation hiérarchique de scènes qui repose sur un arbre de construction dont les feuilles sont des *objets environnementaux*, des objets qui génèrent eux mêmes leurs décorations géométriques. Nous présentons ensuite un formalisme implicite pour définir l'environnement, qui contient l'ensemble des informations guidant la génération des décorations. Finalement, nous détaillons quatre méthodes de génération procédurale pour la création des décorations géométriques (neige, glace, herbes, feuilles) des *objets environnementaux*.

Details modeling in procedural worlds

Procedural modeling of virtual worlds is an active research field in computer science. A large amount of methods have been published in this field : modeling of plants, trees, buildings, cities or terrains, and aging and weathering simulations. In this context, changes of appearance are a very active research field too, due to the way they impact the realism of produced virtual scenes. This research focuses on a procedural method that can represent the changes of appearance as geometrical decorations (snowfall, ice growth, leaves deposits, etc.) on very big scenes with a high level of details.

We first propose a hierarchical scene design based on a construction tree whose leaves are *environmental objects*, a new kind of objects that generate their own geometrical decorations. We then present an implicit formalism to define the environment that contains all the information needed to guide decorations generation. Finally, we detail four procedural methods for generating the geometrical decorations (snow, ice, grass, leaves) of the *environmental objects*.