

Changement d'apparence au cours des saisons

F. Grosbellet¹, A. Peytavie², E. Guérin², E. Galin², S. Mérillou¹ and B. Benes³

¹XLIM - UMR CNRS 7252, Université de Limoges, France

²LIRIS, CNRS UMR 5205, Université de Lyon, France

³Purdue University, USA

Résumé

Nous proposons une nouvelle approche procédurale pour la création et l'édition de grandes scènes composées de nombreux objets. L'apparence des objets peut être modifiée par des facteurs environnementaux tels que la neige, les feuilles et la glace. Une scène est représentée comme un ensemble d'objets procéduraux qui interagissent avec leur environnement. L'environnement est défini comme une combinaison de champs de potentiel, comme des champs de température ou d'occlusion, qui permettent de changer l'apparence des objets au cours du temps. Les caractéristiques saisonnières sont très vite générées en changeant simplement les variables définissant l'environnement. Grâce à ce système, la création d'une scène est simple et rapide, interactive et les changements d'apparence selon la saison de la scène sont générés en quelques secondes. Dans notre implémentation, une scène complexe de plusieurs milliers de mètres carré comportant des détails de moins d'un centimètre est générée en quelques secondes seulement.

1. Introduction

Les modifications d'apparence liées au vieillissement des matériaux ainsi que les variations au cours des saisons sont très importantes pour le réalisme de scènes virtuelles. Ces modifications ont un effet sur la géométrie des objets qui composent la scène ainsi que sur leur rendu. Dans le monde réel, ces changements sont le résultat de processus naturels complexes et dépendants d'un nombre très important de paramètres. Reproduire de tels phénomènes en synthèse d'images reste un verrou scientifique et technique majeur.

La plupart des précédents travaux existants relatifs à la modélisation de changements d'apparence sont soit des méthodes fonctionnant à partir d'exemples, des modèles procéduraux ou des simulations. Pour reproduire ces effets, les utilisateurs définissent les zones affectées, typiquement en peignant sur la surface de l'objet.

Ces méthodes souffrent de deux limitations importantes : l'apparence des éléments constituant la scène est définie par phénomène, les interactions complexes des différents éléments de la scène ne sont pas prises en compte, et les techniques existantes ne permettent pas le passage à l'échelle. Par exemple, une pierre peut être dégradée avec une grande précision, et les méthodes de calcul de neige peuvent être appliquées à de grandes scènes (1 km, précision 1m). Cependant, les interactions ne sont pas prises en compte par ces méthodes. Il est difficile de reproduire de grandes scènes qui prennent en compte différents phénomènes en utilisant des méthodes isolées.

L'observation clef de notre travail est que les changements

saisonniers peuvent être pris en compte par objet plutôt que globalement, car différents objets vont réagir différemment aux changements saisonniers.

En estimant ces changements d'une manière uniforme, utiliser des algorithmes spécifiques à chaque type d'objets plutôt que des algorithmes génériques permet d'apporter des améliorations au résultat final et d'améliorer fortement les performances, comparées à celles des simulations très coûteuses en temps de calcul. La seconde observation est que les objets s'influencent mutuellement. Si les comportements sont spécifiés par objet, une simulation de l'échange de telles informations peut être exprimée facilement.

Pour parvenir, à gérer ce problème d'échelle, nous introduisons le concept des objets réagissant aux paramètres de l'environnement. Ces objets ne définissent pas seulement leur géométrie, mais décrivent également la façon dont leur apparence réagit à l'environnement. Nos objets communiquent de manière bidirectionnelle avec l'environnement : ils sont affectés par cet environnement, mais peuvent également le modifier. Dans notre système, une scène est contrôlée localement, par la définition du comportement des objets, et globalement, en modifiant l'environnement. Nous illustrons notre approche en calculant trois phénomènes : les tapis de neige, de feuilles mortes et de stalactites de glace. La figure Figure 1 montre une scène complexe créée en utilisant notre approche. La scène est composée de plusieurs centaines d'objets paramétrés ; et l'édition des différentes saisons de la scène ne prend que quelques minutes. Une fois entièrement paramétrée, la scène peut changer d'apparence



Figure 1: Une grande scène créée de manière interactive en combinant des primitives réagissant à l'environnement. L'apparence de la scène varie au cours des saisons, de l'hiver (à gauche) à l'automne (à droite) en modifiant quelques paramètres de contrôle.

entre l'été, l'automne et l'hiver, en seulement quelques secondes : modifier les paramètres globaux de la scène est suffisant.

2. État de l'art

Parmi le grand nombre d'articles publiés dans le domaine du vieillissement et du changement d'apparence, nous nous focalisons sur les méthodes liées à la neige, les chutes de feuilles et la formation de glace. Pour une description plus détaillée des méthodes liées au vieillissement, à l'érosion, et aux changements d'apparence, nous renvoyons le lecteur à l'étude [MG08] et au livre [DRS07].

Neige Les méthodes traitant de la neige se regroupent en trois classes principales : les méthodes manuelles, les méthodes à base de simulation de particules et les méthodes utilisant des techniques de déplacement de surface *ad hoc*. [NIDN97] propose une méthode manuelle où l'utilisateur spécifie la répartition de la neige dans la scène et le rendu est effectué par un système de Metaballs. Fearing *et al.* utilisent des particules pour représenter de grands ensembles de flocons, et testent la stabilité de la surface de neige créée pour obtenir une surface réaliste dans [Fea00]. Plusieurs travaux se basent sur les propriétés dynamiques de la neige, comme le transport de la neige par le vent [FO02] et l'animation des flocons de neige [LZK*04]. Une autre méthode permettant de simuler une grande quantité de particules [PTS99] se base sur le déplacement de surfaces et l'utilisation de textures spécifiques pour couvrir de grandes scènes de neige. La méthode proposée dans [FB07] génère des scènes similaires en se basant sur le principe de l'occlusion. Plus récemment, un modèle à base de voxels, utilisant les propriétés thermiques des matériaux et simulant de grandes scènes, a été introduit dans [MG*10]. Muraoka *et al.* [MC00] ont utilisé les propriétés physiques de la neige et de l'eau à l'échelle microscopique pour modéliser la chute ainsi que la fonte de la neige et la surface résultante. Les modèles présentés dans [FG09, FG11] se basent sur la construction de piles de matière pour modéliser la distribution de la neige et produire des résultats complexes tels que les ponts de neige.

Feuilles [MMPP03] ont proposé une méthode orientée image pour la modélisation de feuilles. Peyrat *et al.* ont étendu cette approche en simulant le vieillissement des

feuilles dans [PTMG08]. Récemment, Jeong *et al.* [JPK13] ont présenté une méthode pour produire des feuilles automnales réalistes. La distribution des feuilles dans une scène résulte d'interactions dynamiques complexes, combinant la chute des feuilles à leur transport par le vent et leur empilement sur le sol. Des méthodes réalistes ont été introduites [WH91, WZF*03] pour animer le transport des feuilles par le vent. Cependant, ces méthodes sont coûteuses en temps de calcul et peu adaptées à la simulation de phénomènes d'accumulation. Ces méthodes ont été simplifiées par Desbenoit *et al.* [DGAG06], qui ont ainsi modélisé la distribution de centaines de feuilles sur le sol. Les trajectoires complexes ont été approximées par des modèles de mouvement, comme le flottement ou la chute en spirale, et de larges piles de feuilles ont été créées en utilisant une simple méthode de détection de collision et de stabilisation.

Glace Simuler de grandes quantités de glace est une tâche difficile dû à la grande variété d'échelles concernée. Une approche consiste à utiliser un modèle aléatoire de croissance de glace, où les gouttes d'eau se déplacent sur une surface gelée et on a une probabilité de geler à leur tour [KG93]. Kim *et al.* [KL03, KHL04] ont présenté des algorithmes de formation de glace à petite échelle sur des objets. Kim *et al.* [KAL06] ont introduit une méthode physique pour simuler la croissance de stalactites, en approximant la solidification des couches d'eau par un problème de Stefan.

La plupart de ces travaux ne traite qu'un phénomène isolé, à une échelle soit globale soit locale, et aucun ne permet de traiter de grandes scènes avec un niveau de précision élevé. Nous introduisons une approche unifiée qui définit les interactions mutuelles de nombreux objets et de l'environnement à travers le comportement spécifique à chaque objet. De plus, nous proposons une solution qui résout les problèmes d'échelle en créant des objets réactifs à l'environnement dont la géométrie, l'apparence, la couche de neige, de feuilles et de glace s'adaptent à l'environnement.

3. Aperçu de la méthode

Dans cette section nous présentons l'architecture de notre approche et de ses composants clés.

Définition des objets La scène est un ensemble d'objets qui réagissent dynamiquement aux paramètres de leur en-



Figure 2: Aperçu des différentes étapes d'une scène modélisée avec notre framework.

vironnement. Leurs propriétés internes et la façon dont ils réagissent à leur contexte est spécifique à chaque type d'objet.

Cela nous permet de calculer efficacement les changements d'apparence et de générer de grandes scènes avec un niveau de détails élevé, et ceci en temps interactif. Tout au long de cet article, un objet faisant partie de la scène sera noté \mathcal{O} . Les surfaces de neige, les feuilles et les stalactites seront notées respectivement \mathcal{P} , \mathcal{L} et \mathcal{I} .

Création des effets saisonniers L'environnement est décrit comme un ensemble de champs scalaires qui peuvent être définis soit par l'utilisateur, soit générés par une simulation. Nous définissons un phénomène (neige, feuille ou glace) interactivement par sa région d'influence et son intensité. Les champs scalaires correspondants caractérisent respectivement la quantité de neige tombée, la densité des feuilles et la quantité de glace formée. Les champs de neige, de feuilles et de glace sont notés respectivement $s(\mathbf{p})$, $\lambda(\mathbf{p})$, et $\pi(\mathbf{p})$.

Les champs scalaires sont définis en combinant des fonctions de potentiel à support compact générées par des primitives à squelette, comme décrit dans le modèle BlobTree [WGG99].

Les fonctions des primitives, notées f_i , sont définies par $f_i = g_i \circ d_i$, où g_i définit une fonction à support compact et d_i définit une fonction de distance à un squelette. Une grande variété d'effets peut être obtenue en ajustant la taille, la forme et l'intensité des primitives à squelette.

Contrôle de l'environnement L'utilisateur définit les propriétés locales et globales de l'environnement. L'occlusion et la température sont définies par des champs scalaires qui augmentent ou diminuent l'intensité d'un phénomène.

Le champ de température est noté $\theta(\mathbf{p})$, et celui d'occlusion $\alpha(\mathbf{p})$. Le contrôle de l'environnement peut être *global* ou *local*. Les champs scalaires globaux définissent les paramètres liés à l'intégralité de la scène, alors que des champs scalaires locaux permettent un contrôle plus précis et plus ciblé dans une région de l'espace.

Workflow La création et l'édition interactive de la scène sont réalisées en ajoutant ou en supprimant des objets et en éditant les champs qui influencent leur apparence. Ces étapes peuvent être réalisées dans un ordre arbitraire : il est possible d'éditer, de rajouter ou de supprimer des objets ou des éléments de contrôle à n'importe quel moment.

Dans la Figure 2, l'utilisateur définit la scène initiale en assemblant une primitive de trottoir et de route. Il ajoute

un arbre, une poubelle ou un banc (Figure 2.1). Par défaut, les paramètres *globaux* de l'environnement correspondent à l'été : l'arbre génère donc des feuilles vertes. Dans la Figure 2.2 les paramètres *globaux* de l'environnement correspondent à l'automne. L'arbre génère un large volume sphérique de feuilles mortes qui modifie l'apparence de la route et du trottoir. La Figure 2.3 montre un exemple de contrôle utilisateur : deux petits volumes de feuilles ont été ajoutés à la scène pour créer deux petites piles de feuilles sur le trottoir. La Figure 2.4 correspond à la même scène, mais en hiver. En changeant les paramètres *globaux* pour les faire correspondre à l'hiver, les primitives de route, de trottoir, de banc et d'arbre génèrent automatiquement leur couche de neige. Dans la dernière étape (Figure 2.5) des traces de pas sont introduites dans la neige, en utilisant un ensemble de champs de contrôle *locaux*.

4. Effets saisonniers

Notre travail porte sur trois phénomènes climatiques importants qui ont un fort impact visuel : la neige, les feuilles mortes et la glace. Dans les paragraphes suivants nous décrivons le fonctionnement de chacun de ces effets en détail.

4.1. Neige

La neige est gérée comme un maillage auto-généré qui s'adapte à l'échelle et à la géométrie de l'objet qu'elle recouvre. Notre but est d'obtenir un calcul rapide de la couche de neige en conservant une édition interactive.

Notre approche consiste à couvrir la surface d'un objet avec des *patches de neige*. Un patch de neige est noté \mathcal{P} et correspond à un maillage qui peut être défini par un utilisateur ou être généré procéduralement. Les sommets de ce maillage sont déplacés par rapport à la surface de l'objet d'un seuil qui correspond à la quantité de neige. Ce seuil est également affecté par l'environnement (occlusion et température) comme décrit dans la section 5.

4.1.1. Patches de neige

Un patch de neige \mathcal{P} est généré depuis le maillage initial de l'objet $\mathcal{M}(0)$ qui est le support de la couche de neige correspondante. Les sommets du patch de neige sont référencés comme \mathbf{p}_k , et ils ont un vecteur directeur \mathbf{v}_k associé (Figure 3). Le vecteur directeur \mathbf{v}_k est la plupart du temps vertical \mathbf{z} , mais les vecteurs \mathbf{v}_k associé aux vertex du bord peuvent être légèrement inclinés pour permettre la création de surplombs de neige.

Un patch de neige a une fonction d'élévation $\delta_{\mathcal{P}}(\mathbf{p}_k, s(\mathbf{p}_k))$ associée qui définit le déplacement

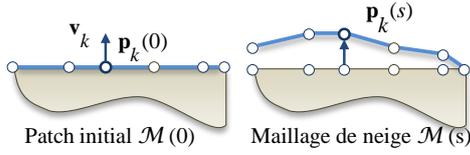


Figure 3: Définition d'un patch de neige.

des sommets de son maillage en fonction de la quantité de neige tombée $s(\mathbf{p}_k)$. Les positions des sommets du maillage sont calculées de la façon suivante :

$$\mathbf{p}_k(s) = \mathbf{p}_k(0) + \delta_{\mathcal{P}}(\mathbf{p}_k(0), s(\mathbf{p}_k(0))) \mathbf{v}_k$$

Nous définissons deux différentes catégories de patches de neige : les patches *génériques* et les patches *construits par la géométrie*. Les patches *génériques* sont des maillages procéduraux caractérisés par un ensemble de paramètres définissant leur forme et leur niveau de subdivision (Figure 4).

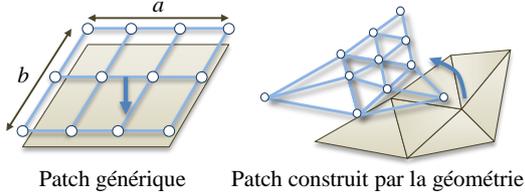


Figure 4: Un patch générique (à gauche) et un exemple de patch construit à partir de la géométrie (à droite).

Les patches *construits à partir de la géométrie* sont dérivés du maillage initial de l'objet $\mathcal{M}(0)$. Cette approche est adaptée pour la création de patches de neige complexes liés à des objets dont la géométrie est complexe, comme les arbres par exemple (Figure 5).

Les méthodes utilisant un découpage de l'espace en grille régulière [FG09], rencontrent des problèmes liés au positionnement des objets sur la grille, alors que le fonctionnement de notre méthode résout directement ce type de problèmes.

En faisant correspondre les sommets d'un patch de neige aux sommets de l'objet supportant ce patch, nous évitons les fissures qui pourraient apparaître entre la surface de neige et la géométrie de l'objet. De plus, en utilisant notre approche, les patches de neige peuvent être raffinés progressivement afin d'obtenir un maillage plus précis. Cela nous permet de traiter de grandes scènes efficacement, et de générer facilement différents niveaux de détails.

Les patches de neige sont automatiquement générés à partir des paramètres de l'objet \mathcal{O} auxquels ils correspondent. Un exemple d'adaptation est montré dans la Figure 6 : une fenêtre redimensionnée régénère automatiquement sa couche de neige.

Un objet peut comporter plusieurs patches de neige génériques : la Figure 5 montre un banc couvert de neige.

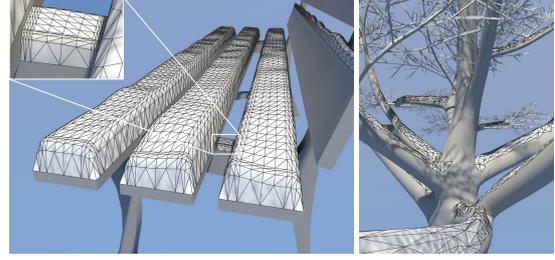


Figure 5: Des patches de neige génériques ont été utilisés pour la génération neige sur le banc (gauche), alors que des patches de neige ont été extraits de la géométrie de l'arbre (droite).



Figure 6: Les objets paramétrés génèrent automatiquement leur apparence, comme on peut le voir sur cette fenêtre redimensionnée.

Plusieurs patches de neige génériques quadrilatéraux ont été combinés et instanciés pour créer la couche de neige finale.

4.1.2. Calcul de l'élévation de la neige

La fonction $\delta_{\mathcal{P}}$ est fortement corrélée à la géométrie locale de l'objet \mathcal{O} . Dans notre implémentation, l'élévation de la neige peut être soit définie procéduralement en utilisant différentes classes de fonctions paramétrées, soit calculée à partir d'enregistrements de différents hauteurs de neige successives obtenus à partir d'une simulation.

Fonctions d'élévation procédurale La fonction d'élévation $\delta_{\mathcal{P}}$ est définie comme une combinaison de fonctions paramétrées pour les patches de neige *génériques* associés à des objets géométriques simples. Cette approche est particulièrement efficace pour des primitives simples comme des boîtes, des cylindres, des sphères ou des surfaces de révolution, qui sont fréquemment utilisées dans les modèles architecturaux. Les patches de neige *génériques* sont entièrement procéduraux et ne prennent que très peu de place en mémoire. Nous définissons des fonctions différentes pour chaque type de patch de neige. Nous utilisons $\delta_{\mathcal{P}}(\mathbf{p}) = s(\mathbf{p}) \cdot g(d(\mathbf{p}))$, où $d(\mathbf{p})$ correspond à une fonction de distance entre le point du patch de neige et le bord du patch, et g à la courbe de profil de la neige définie comme :

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

Le paramètre d_0 contrôle la pente de la neige aux bords de l'objet.

Simulation : Il est souvent difficile de faire correspondre des patchs de neige *génériques* sur des objets dont la géométrie est complexe (tels que des statues ou des arbres) qui comportent de nombreuses parties concaves avec des phénomènes d'occlusion interne. Pour répondre à ce problème, nous définissons la fonction d'élévation $\delta_{\mathcal{P}}(\mathbf{p}_k, s)$ automatiquement en réalisant une simulation avec différentes hauteur de neige successives sur l'objet et en enregistrant la position des sommets \mathbf{p}_k du patch de neige pour un ensemble donné de hauteurs de neige $s_i = i/(n-1), i \in [0, n-1]$. La fonction de déplacement finale $\delta_{\mathcal{P}}(\mathbf{p}_k, s)$ est obtenue en interpolant ces échantillons.

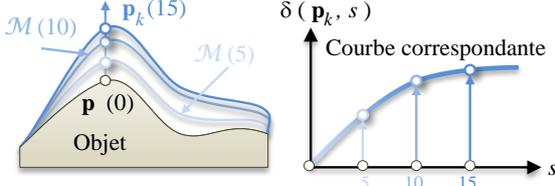


Figure 7: Fonction d'élévation calculée à partir d'une simulation physique de chute de neige.

L'un des désavantages liés à l'enregistrement de simulations, comparé au patchs de neige *génériques*, est leur coût en mémoire. Nous avons observé que dans de nombreux cas, les échantillons obtenus suivent des modèles simples (comme une croissance linéaire) avant d'atteindre une limite (Figure 7). Pour résoudre ce problème de mémoire dans ces cas, nous utilisons une fonction paramétrée dont nous ne stockons que les paramètres de contrôle, ce qui nous permet d'économiser de la mémoire et d'accélérer le calcul d'élévation. Dans notre système, nous utilisons la fonction suivante :

$$g(s) = a \frac{(s/s_0)}{\sqrt{1 + (s/s_0)^2}}$$

Le paramètre a contrôle la hauteur de neige maximum et s_0 définit la transition entre la croissance linéaire et la limite maximale.

4.2. Feuilles

La distribution des feuilles est définie par le champ scalaire de feuilles noté $\lambda(p)$. Nous présentons une méthode en deux étapes pour la génération de tas de feuilles dans une scène. Premièrement, nous construisons un ensemble de feuilles candidates pour chaque classe d'objets. Ensuite, nous instancions les feuilles en fonction de la valeur du champ scalaire de feuilles et des paramètres de l'environnement.

4.2.1. Tas de feuilles

Nous définissons un *tas de feuilles* comme un ensemble de modèles de feuilles $\{\mathcal{L}_i, i \in [0, n-1]\}$ liés à un ensemble de positions dans l'espace. Chaque modèle de feuille (Figure 8) $\mathcal{L}_i = \{\mathbf{p}_i, \mathbf{R}_i, d_i, a_i\}$ est décrit par sa position \mathbf{p}_i , sa matrice de rotation \mathbf{R}_i , sa distance d_i à la géométrie de l'objet \mathcal{O} et son index a_i correspondant à un type de modèle géométrique.

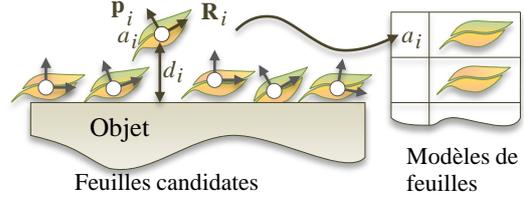


Figure 8: Attributs d'un modèle de feuille.

Nous utilisons trois techniques complémentaires pour générer les tas de feuilles : édition manuelle, simulation physique liée à un type d'objets particulier, et approche procédurale fonctionnant avec un pavage aperiodique de blocs de feuilles en trois dimensions.

Édition manuelle La structure d'un tas de feuille peut être éditée manuellement. Cela permet à l'utilisateur de placer facilement des feuilles candidates à la surface de l'objet. En pratique, cette technique est utile pour positionner des feuilles sur des objets concaves. Elle permet de reproduire facilement des cas réels où les feuilles déplacées par le vent s'accrochent à des surfaces rugueuses ou se retrouvent coincées ou épinglées sur un objet. Ce type de cas est extrêmement compliqué à reproduire avec une simulation. Un exemple est présenté sur la partie droite de la figure Figure 9.

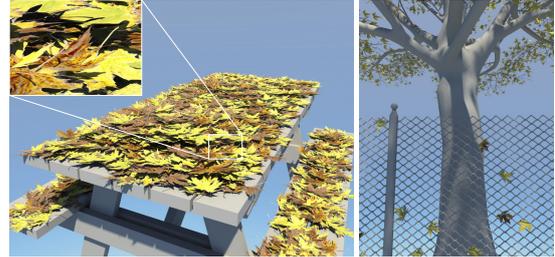


Figure 9: Un tas de feuilles quadrilatéral dont la distribution de feuilles est procédurale génère des feuilles sur une table (à gauche) tandis que des positions spécifiques ont été éditées manuellement afin de permettre l'instanciation de feuilles accrochées à une barrière (à droite).

Simulation L'édition manuelle est fastidieuse et ne convient pas à la modélisation de grands tas de feuilles. Pour ce type de cas, nous utilisons la simulation de chute de feuilles présentée dans [DGAG06]. Ce processus enregistre les positions \mathbf{p}_i et les orientations \mathbf{R}_i des feuilles tombées dans la structure du tas de feuilles attaché à l'objet.

Pavage Le pavage aperiodique en trois dimensions est une méthode rapide et efficace pour générer des tas de feuilles sur un ensemble d'objets. Notre technique est basée sur la méthode de génération de pierres présentée dans [PGGM09] et est constituée de deux étapes. Tout d'abord, un ensemble de cellules cubiques qui contiennent des couches de feuilles réparties de manière aperiodique est généré. Ensuite, nous définissons le tas de feuilles attaché à un objet en le plongeant dans les cellules, et en conservant uniquement les

feuilles situées dans un volume de contrôle qui définit la forme du tas (Figure 10).

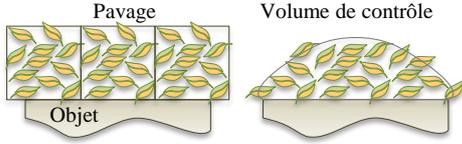


Figure 10: Pavage apériodique utilisé pour créer efficacement des piles de feuilles.



Figure 11: Constitution du pavage par couches.

Les cellules cubiques remplies de feuilles sont créées en incrémentant différentes couches de feuilles (Figure 11). Chaque couche de feuilles est créée en utilisant une distribution de Poisson avec un rayon r . Le rayon doit être légèrement plus grand que le rayon moyen du cylindre englobant des feuilles afin d'éviter les intersections entre feuilles.

4.2.2. Instantiation des feuilles

Les tas de feuilles attachés à leur objet \mathcal{O} définissent un ensemble de feuilles candidates pour chaque objet de la scène. Une feuille \mathcal{L}_i est instanciée si la valeur de la fonction du champ de feuille $\lambda(\mathbf{p}_i)$ est plus grand que la distance d_i (Figure 12). Le sous ensemble instancié est défini de la manière suivante :

$$\mathcal{L} = \{\mathcal{L}_i \mid \lambda(\mathbf{p}_i) < d_i\}$$

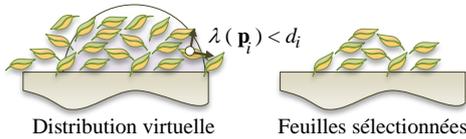


Figure 12: Lors de la phase d'instanciation des feuilles seules les feuilles dont $\lambda(\mathbf{p}_i) < d_i$ sont instanciées.

4.3. Stalactites de glace

Un ensemble de stalactites de glace, noté $\{\mathcal{I}_j, j \in [0, m - 1]\}$, peut être lié à un objet. Individuellement, chaque stalactite est décrite par sa position d'attache $\mathcal{I}_j = \{\mathbf{p}_j\}$ et par la fonction de glace $\pi(\mathbf{p})$ qui donne la quantité de glace au point \mathbf{p} . A chaque fois que la fonction $\pi(\mathbf{p}_j)$ est strictement positive, une stalactite de glace sera générée (Figure 13) :

$$\mathcal{I} = \{\mathcal{I}_j \mid \pi(\mathbf{p}_j) > 0\}$$

La position d'une stalactite peut être définie manuellement ou automatiquement le long des arêtes d'un objet. La géométrie des stalactites peut être créée par un artiste, par une méthode de simulation [KAL06], ou générée procéduralement. Nous utilisons une représentation procédurale basée sur le BlobTree présenté dans [WGG99].

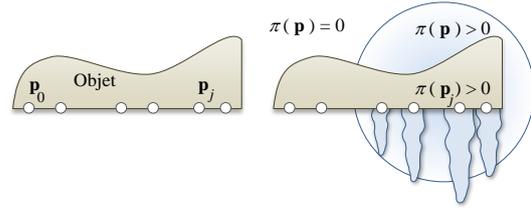


Figure 13: Ensemble des points d'accroche \mathbf{p}_j liés à un objet (gauche). La sélection et les dimensions des stalactites dépendent de la valeur de $\pi(\mathbf{p}_j)$ (droite).

Génération à base de surfaces implicites Les propriétés géométriques des stalactites sont calculées à partir de la valeur $\pi(\mathbf{p}_j)$. Le rayon de la base r_j et la longueur l_j des stalactites sont proportionnels à $\pi(\mathbf{p}_j)$.

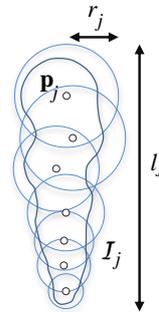


Figure 14: Modélisation des stalactites avec une surface implicite.

Une liste de primitives sphériques est générée le long de l'axe des stalactites (Figure 14) en descendant à partir de sa base. Le rayon des sphères décroît continuellement, comme l'espace qui sépare deux sphères successives. Une perturbation aléatoire horizontale est utilisée pour améliorer l'apparence finale des stalactites. Les primitives à squelette ponctuel générées sont combinées dans un même modèle implicite avant de générer la surface maillée.

Le maillage final des stalactites est produit avec une technique de polygonisation (marching cube). La précision de la grille utilisée dépend des caractéristiques géométriques de la stalactite. Quand deux stalactites sont trop proches, un seul maillage est créé en combinant la représentation implicite de ces deux stalactites (Figure 15 et 16).



Figure 15: Stalactites de glace formées sur une fontaine.

Instanciation Afin d'éviter la génération de trop nombreuses stalactites, nous instancions les stalactites qui ont les mêmes propriétés (rayon de la base et longueur). Ces modèles instanciés peuvent être pré-calculés afin de réduire le temps de calcul de la scène.



Figure 16: Stalactites de glace sur un lampadaire.

5. Environnement

Les effets environnementaux sont initialement produits et contrôlés par les fonctions de champ de neige, de feuilles et de glace qui caractérisent la répartition et l'importance de chacun de ces phénomènes. De plus, ils peuvent être contrôlés par deux fonctions environnementales, l'*occlusion* et la *température*, qui influencent la surface de neige et la distribution des feuilles et des stalactites de glace.

Il est possible de calculer la température et l'occlusion en utilisant respectivement une simulation de transfert thermique et la géométrie des primitives de la scène. Dans notre système, nous proposons d'utiliser une méthode de calcul plus rapide basée sur des primitives implicites, afin de pouvoir créer facilement et rapidement des champs complexes. Bien que cette approche ne soit pas exacte, elle fournit à l'utilisateur un outil efficace et intuitif qui permet un contrôle précis pour créer des scènes complexes.

5.1. Définition des fonctions environnementales

Nous calculons la température et l'occlusion en combinant des primitives à squelette avec des fonctions à support compact. Les primitives sont définies comme $\alpha_i = g_i \circ d_i$, où $g_i : \mathbf{R}_+ \rightarrow [0, 1]$ est une fonction à support compact.

Occlusion La fonction d'occlusion α est définie en multipliant la valeur des différentes primitives d'occlusion α_i :

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

Le potentiel des primitives $\alpha_i(\mathbf{p})$ se situe dans l'intervalle $[0, 1]$. En multipliant les potentiels plutôt que de les sommer comme dans les modèles de surfaces implicites traditionnels, nous obtenons une combinaison plus naturelle et intuitive des primitives d'occlusion.

Température La fonction de température θ est définie en sommant les contributions des différentes émettrices de chaleur :

$$\theta(\mathbf{p}) = \sum_{i=1}^{i=n} \theta_i(\mathbf{p})$$

De cette manière, les différents champs de température ont un effet cumulatif. Notons que ce modèle permet d'utiliser des champs de température négatifs pour ajouter de la neige au lieu de la faire fondre (Section 6).

5.2. Influence sur la neige, les feuilles et la glace

Les effets environnementaux ont un effet direct sur les trois phénomènes que nous avons implémentés.

La **neige** est affectée par les champs de température et d'occlusion (Figure 17) :

$$\lambda(s) = \mathbf{p}_k(0) + (\alpha(\mathbf{p}_k) \cdot \delta_{\mathcal{P}}(\mathbf{p}_k, s(\mathbf{p}_k)) - \theta(\mathbf{p}_k)) \mathbf{v}_k$$

Le terme $\alpha(\mathbf{p})$ est la valeur du champ d'occlusion et opère comme un facteur multiplicatif qui limite l'accumulation de neige. Le terme $\theta(\mathbf{p})$ correspond à une source de chaleur et opère comme un terme soustractif qui réduit l'épaisseur de la couche de neige.

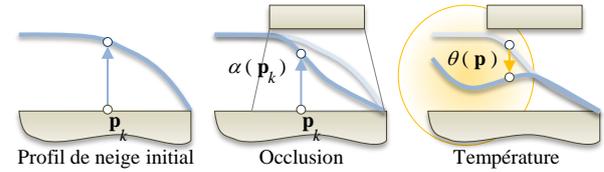


Figure 17: Les effets de l'environnement sont définis par les champs d'occlusion (au centre) et de température (à droite) et affecte la fonction initiale de déplacement de la neige $\delta_{\mathcal{P}}(\mathbf{p}_k, s)$.

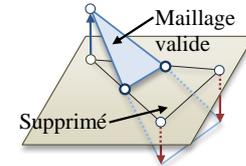


Figure 18: Raffinement local du maillage.

Le calcul de $\delta(\mathbf{p})$ peut parfois aboutir à une valeur de déplacement négative à cause d'une température élevée. Dans ce cas, la neige ne doit pas être présente au point \mathbf{p} . Nous raffinons alors localement le maillage de la surface de la neige pour créer un maillage adapté (Figure 18).

Les feuilles ont une distribution dépendante des caractéristiques de l'environnement et prennent en compte les champs d'occlusion de la manière suivante :

$$\tilde{\mathcal{L}} = \{\mathcal{L}_i \mid \alpha(\mathbf{p}_i) \cdot \lambda(\mathbf{p}_i) < d_i\}$$

L'occlusion diminue virtuellement le champ de feuilles et diminue donc localement la hauteur des tas de feuilles.

Les stalactites de glace sont affectées par les sources de chaleur qui modifient leurs caractéristiques. Le champ de température $\theta(\mathbf{p}_j)$ est pris en compte de la façon suivante pour diminuer les champs de glace :

$$\tilde{\mathcal{I}} = \{\mathcal{I}_j \mid \alpha(\mathbf{p}_j) \cdot \pi(\mathbf{p}_j) - \theta(\mathbf{p}_j) > 0\}$$

Ainsi, une stalactite n'apparaîtra au point d'échantillonnage \mathbf{p}_j que si $\alpha(\mathbf{p}_j) \cdot \pi(\mathbf{p}_j) - \theta(\mathbf{p}_j) > 0$. De plus, les paramètres de longueur et de rayon sont modifiés par l'accessibilité et la température pour obtenir des formes plus petites en fonction des champs de potentiel de l'environnement.



Figure 19: Exemple de contrôle utilisateur : des empreintes et un tas de neige peuvent être ajoutés dans la scène.



Figure 20: Piles de feuilles et feuilles accumulées par le vent contrôlées par l'utilisateur.

6. Résultats

Nous avons implémenté notre méthode en C++ et tous les développements ont été effectués sur un ordinateur équipé d'un Intel[®] Core i7 cadencé à 3GHz avec 16GB de RAM. Les données produites par notre système ont été directement envoyées vers MentalRay[®] pour produire les images photoréalistes présentées tout au long de cet article.

Contrôle Notre approche permet de contrôler interactivement la création de scènes complexes intégrant différents phénomènes naturels caractéristiques des saisons tels que des piles de feuilles en automne, des tas de neige, des traces et des empreintes dans la neige en hiver. Notre méthode permet également de définir simplement des zones où la neige fond ou se transforme en glace sans recourir à des simulations coûteuses en temps de calcul.

La Figure 19 montre une scène où des traces de roues et de pas ont été éditées à l'aide de champs de température et d'occlusion. La Figure 20 montre la distribution des feuilles et la création d'empilements.

Une modélisation intuitive de la scène a été réalisée en ajoutant des champs de température et d'occlusion (Figure 19). Le système de contrôle permet une édition similaire de la distribution des feuilles, empilements compris (Figure 20).

Les fonctions de température et d'occlusion jouent un rôle important dans le réalisme global de la scène. Dans les cas que nous avons choisis, elles fournissent des contrôles permettant également d'augmenter l'épaisseur de neige en utilisant des champs de température négative, produisant alors des tas de neige (Figure 19).

Les primitives d'occlusion peuvent être utilisées pour créer des détails ou ajouter des éléments spécifiques à une scène virtuelle. Les empreintes dans la neige visibles dans la Figure 19 ont été créées à partir d'une primitive à squelette combinée à une carte de profondeur et interprétée comme un champ d'occlusion. Réparties à différentes positions, elles reproduisent les traces de pas de piétons.

Échelles Notre approche procédurale peut générer des changements saisonniers sur de grandes scènes qui comportent un haut niveau de détails. Ces résultats seraient difficiles à obtenir avec les techniques existantes qui reposent sur une décomposition de l'espace en grille régulière.

Notre approche intègre très naturellement la gestion de la multi-résolution et permet de visualiser la scène à différents niveaux de précision, tout en utilisant des algorithmes de sélection partielle des éléments de la scène ou des stratégies basées sur la gestion des ressources. Les exemples présentés dans les Figure 21 et 22 montrent différentes scènes de ville où la précision des objets varie de 20cm (primitives de route et de trottoir) à 5 cm (fenêtre et escalier) et jusqu'à 1 cm (lampadaire) (voir également Table 1).

Scène	Durée	Triangles	Taille	Détails
Contrôle	6.5	3.1	6 × 6	1
Parc	3.5	22.0	115 × 90	1 – 20
Paris	22.5	15.8	33 × 100	1 – 10

Table 1: Statistiques pour la génération de la neige dans différentes scènes. Les durées sont exprimées en secondes, le nombre de triangles en millions, la taille en m² et les détails en cm.

Performances Les primitives paramétrées permettent facilement le passage à l'échelle et la gestion de grandes scènes. Des exemples de plusieurs milliers de mètres carrés (Figure 21 et Figure 22) sont générés en quelques secondes seulement (Table 1 et Table 2). Notre approche phénoménologique contrôlant la distribution de la neige, des feuilles et de stalactites de glace ne repose pas sur une simulation physique complexe et permet une édition interactive.

Scène	Durée	Feuilles	Taille
Contrôle	0.1	6.3	6 × 6
Parc	0.6	78.7	115 × 90
Paris	0.6	111	33 × 100

Table 2: Statistiques pour la génération de feuilles dans différentes scènes. Les durées sont exprimées en secondes, le nombre de feuilles en milliers et la taille en m².

7. Conclusion

Nous avons présenté une nouvelle méthode pour l'édition de grandes scènes qui sont affectées par des facteurs environ-



Figure 21: Différents points de vue d'un parc à Chicago en automne et en hiver.

nementaux tels que la neige, les feuilles et la glace. Contrairement aux méthodes génériques existantes qui traitent l'ensemble des objets de manière homogène, notre méthode permet l'utilisation de paramètres spécifiques à chaque objet pour optimiser les traitements. Cela permet une gestion efficace des différents phénomènes au sein d'un modèle unifié.

Afin de fournir plus d'outils de contrôle, nous utilisons des variations d'occlusion et des sources de température pour faire varier l'intensité des phénomènes naturels. Notre méthode s'inspire des champs de potentiels utilisés pour la modélisation de surfaces implicites. Chaque élément de contrôle est une primitive à squelette qui définit son propre champ d'influence. Grâce à notre système, nous sommes capables de générer en quelques secondes différents types de scènes de plusieurs milliers de mètres carrés avec un très haut niveau de détail.

Notre méthode présente cependant quelques limitations. Dans la nature, lorsque la quantité de neige tombée est importante, la couche de neige finale obtenue est une surface lissée qui dissimule les détails des objets. Notre système permet de traiter ces cas, en construisant de nouveaux patches de neige spécifiques pour des ensembles d'objets de manière à gérer une grande quantité de neige, mais cette approche nécessite la définition très précise des propriétés des différents objets, étape relativement longue à réaliser. Une autre limitation évidente de notre méthode est que nos scènes sont actuellement statiques. Un système d'animation augmenterait de manière significative le réalisme.

Notre méthode pourrait être améliorée de différentes façons. Une première amélioration consisterait à combiner notre approche avec des techniques de simulation dynamique, telle que le transport des feuilles par le vent. Une autre amélioration serait de généraliser notre approche à d'autres phénomènes naturels ou saisonniers. Comme notre méthode gère les changements d'apparence saisonniers pour

chaque type d'objet avec des techniques optimisées pour différents effets (neige, feuille et glace), de nombreux algorithmes comme que le vieillissement, l'apparition de fissures ou de fractures, la corrosion, la croissance de lichens ou de mousses pourraient être insérés dans notre système.

Références

- [DGAG06] DESBENOIT B., GALIN E., AKKOCHE S., GROSJEAN J. : Modeling autumn sceneries. In *Eurographics Short Papers* (2006), pp. 107–110.
- [DRS07] DORSEY J., RUSHMEIER H., SILLION F. : *Digital Modeling of Material Appearance*. Computer Graphics. Morgan Kaufmann / Elsevier, 2007.
- [FB07] FOLDES D., BENES B. : Occlusion-based snow accumulation simulation. In *Workshop on Virtual Reality Interactions and Physical Simulation* (2007), pp. 35–41.
- [Fea00] FEARING P. : Computer modelling of fallen snow. In *Proceedings of Siggraph* (2000), pp. 37–46.
- [FG09] FESTENBERG N., GUMHOLD S. : A geometric algorithm for snow distribution in virtual scenes. In *Proceedings of the Fifth Eurographics conference on Natural Phenomena* (2009), pp. 17–25.
- [FG11] FESTENBERG N., GUMHOLD S. : Diffusion based snow cover generation. *Computer Graphics Forum*. Vol. 30, Num. 6 (2011), 1837–1849.
- [FO02] FELDMAN B., O'BRIEN J. : Modeling the accumulation of wind-driven snow. In *Proceedings of Siggraph 2002, Technical Sketch* (2002), p. 218–218.
- [JPK13] JEONG S., PARK S.-H., KIM C.-H. : Simulation of morphology changes in drying leaves. *Computer Graphics Forum*. Vol. 32, Num. 1 (2013), 204–215.
- [KAL06] KIM T., ADALSTEINSSON D., LIN M. : Modeling ice dynamics as a thin-film stefan problem. In *Pro-*



Figure 22: Différents points de vue d'un quartier du style parisien en automne et en hiver.

ceedings of ACM SIGGRAPH/Eurographics symposium on Computer animation (2006), pp. 167–176.

- [KG93] KHARITONSKY D., GONCZAROWSK J. : A physically based model for icicle growth. *The Visual Computer*. Vol. 32, Num. 1 (1993), 88–100.
- [KHL04] KIM T., HENSON M., LIN M. : A hybrid algorithm for modeling ice formation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 305–314.
- [KL03] KIM T., LIN M. : Visual simulation of ice crystal growth. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 86–97.
- [LZK*04] LANGER M., ZHANG L., KLEIN A., BHATIA A., PEREIR J., REKHI D. : A spectral-particle hybrid method for rendering falling snow. In *Proceedings of Eurographics Symposium on Rendering* (2004), pp. 217–226.
- [MC00] MURAOKA K., CHIBA N. : Visual simulation of snowfall, snow cover and snowmelt. In *Proceedings Seventh International Conference on Parallel and Distributed Systems : Workshops* (2000), pp. 187–194.
- [MG08] MÉRILLOU S., GHAZANFARPOUR D. : A survey of aging and weathering phenomena in computer graphics. *Comput. Graph.* Vol. 32, Num. 2 (2008), 159–174.
- [MGG*10] MARÉCHAL N., GUÉRIN E., GALIN E., MÉRILLOU S., MÉRILLOU N. : Heat Transfer Simulation for Modeling Realistic Winter Sceneries. *Computer Graphics Forum*. Vol. 29, Num. 2 (2010), 449–458.
- [MMPP03] MUNDERMANN L., MACMURCHY P., PIVOVAROV J., PRUSINKIEWICZ P. : Modeling lobed leaves. In *Proceedings of the Computer Graphics International Conference* (2003), p. 60–65.
- [NIDN97] NISHITA T., IWASAKI H., DOBASHI Y., NAKAMAE E. : A modeling and rendering method for snow

by using metaballs. *Computer Graphics Forum*. Vol. 16, Num. 3 (1997), 357–364.

- [PGGM09] PEYTAVIE A., GALIN E., GROSJEAN J., MÉRILLOU S. : Procedural generation of rock piles using aperiodic tiling. *Computer Graphic Forum*. Vol. 28, Num. 3 (2009), 1801–1809.
- [PTMG08] PEYRAT A., TERRAZ O., MÉRILLOU S., GALIN E. : Generating vast varieties of realistic leaves with parametric 2Gmap L-Systems. *The Visual Computer*. Vol. 24, Num. 7-9 (2008), 807–816.
- [PTS99] PREMOZE S., THOMPSON W., SHIRLEY P. : Geospecific rendering of alpine terrain. In *Eurographics Workshop on Rendering* (1999), pp. 107 – 118.
- [WGG99] WYVILL B., GUY A., GALIN E. : Extending the csg tree - warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphic Forum*. Vol. 18, Num. 2 (1999), 149–158.
- [WH91] WEJCHERT J., HAUMANN D. : Animation aerodynamics. *ACM SIGGRAPH Computer Graphics*. Vol. 25, Num. 4 (1991), 19–22.
- [WZF*03] WEI X., ZHAO Y., FAN Z., WEI L., YOAKUM-STOVER S., KAUFMAN A. : Blowing in the wind. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer animation* (2003), pp. 75–85.