

Deuxième partie

Modèles d'illumination hiérarchiques et analytiques [MN00]

Modèles d'illumination analytiques et hiérarchiques

La complexité des scènes d'extérieur pose de nombreux problèmes en terme de modélisations et de rendu. Cependant, nous venons de voir que les techniques de modélisation donnent aujourd'hui des résultats réalistes et diversifiés, aux prix de nombreux polygones. Les arbres font partie de cette catégorie d'objets dont la surface n'est pas vraiment définie, ce qui rend les parties internes de leur feuillage potentiellement visibles et éclairées. De plus, le rendu de ce type de scène est extrêmement coûteux, et très sujet à l'aliassage. Nous avons montré dans l'état de l'art que l'utilisation des niveaux de détails permet d'améliorer ce type de problème pour d'autres familles d'objets (*e.g.* herbe, terrain, etc.). Pour les arbres, peu de techniques sont capables de calculer automatiquement des niveaux de détails sans changer l'apparence globale. Nous verrons en 1 comment il est possible de s'appuyer sur la hiérarchie naturelle des arbres pour construire des niveaux de détails.

Les détails géométriques comme les feuilles ou les aiguilles sont trop petits pour être visibles individuellement dès que l'observateur se trouve à quelques dizaines de mètres (*cf.* figure 3.1). Les rameaux eux-mêmes se confondent avec la distance, puis les branches, etc. Il est donc intéressant d'essayer de remplacer les données non distinguables visuellement par une primitive ayant le même comportement photométrique que le groupe de géométrie qu'elle représente. Nous montrerons en 2 que cette idée, couplée à la notion de hiérarchie, est une piste pour la construction de niveaux de détails efficaces.

Lors du rendu, il est important d'utiliser un algorithme tirant partie de la connaissance disponible a priori sur le type de données que l'on souhaite traiter pour limiter l'aliassage et diminuer les coûts au maximum. Nous expliquerons donc en 3 les raisons pour lesquelles nous avons choisi le lancer de cônes, et le critère que nous utilisons pour le choix des niveaux de détails.



FIG. 3.1 – Un exemple de la complexité visuelle d'une forêt. L'objectif est de la représenter autrement que par une forte complexité géométrique, très coûteuse en temps de rendu et génératrice d'aliasage.

1 Niveaux de détails pour les arbres

Au fur et à mesure que l'on s'éloigne d'un arbre, on ne discerne plus les différentes parties : on commence par ne plus distinguer les feuilles, les rameaux puis les branches se mélangent, et pour finir, seule la silhouette de l'arbre reste visible. On ne peut pas non plus supprimer simplement les détails fins : la forêt de la figure 3.1 démunie de ses aiguilles aurait une allure très différente, bien que chaque aiguille occupe moins de 1% d'un pixel. Ce constat plaide en faveur des niveaux de détails et de leur continuité.

1.0.1 Simplification de maillage

L'état de l'art montre que peu de modèles sont capables de générer automatiquement et efficacement des niveaux de détails pour les arbres. Le modèle de Weber et Pen (*cf.* chapitre 2 section 1.2.3) simplifie un modèle géométrique d'arbre en supprimant les branches les moins significatives. Ceci allège le coût de rendu mais au détriment de la conservation de l'illumination et de la transparence générale de l'arbre entre les niveaux. Des effets de saut (*poping*) se font sentir lors du passage d'un niveau à l'autre. Une telle technique n'est pas satisfaisante ; plutôt que de décimer, il est sûrement préférable de regrouper.

1.0.2 Utilisation de la hiérarchie

Les arbres possèdent des propriétés hiérarchiques intéressantes : les feuilles se répètent autour d'un tronc pour former une branche secondaire. Plusieurs branches secondaires se répètent pour former une branche principale et plusieurs branches principales forment l'arbre complet (*cf.* figure

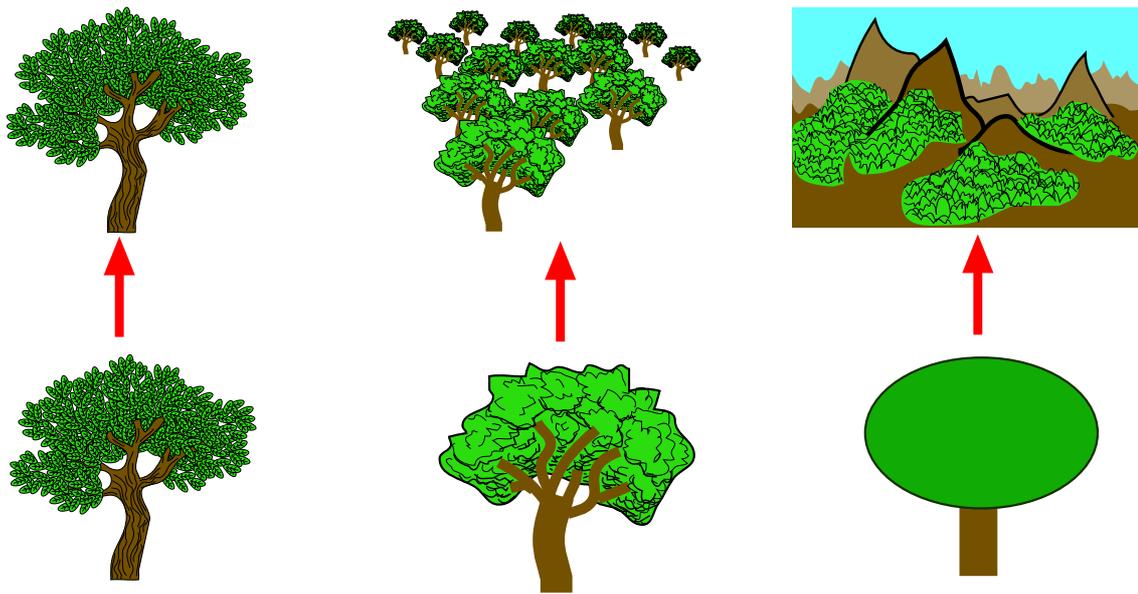


FIG. 3.2 – Différents niveaux de détails.

3.3). De plus, deux branches d'une même famille d'arbres sont assez similaires. Il est même possible d'étendre cette notion de répétition aux paysages puisque deux arbres d'une même famille sont assez semblables.

Ce principe de hiérarchie répétitive peut nous aider pour la construction de niveaux de détails sur deux points :

- la hiérarchie naturelle de l'arbre guidera la hiérarchie de nos niveaux de détails ;
- les structures répétées pourront être factorisées et instanciées, ce qui permet de diminuer le coût mémoire. Pour éviter les effets répétitifs il suffit souvent de changer l'orientation et la taille des objets, voire la palette de couleurs.

Max [MDK99] utilise déjà avec succès ces propriétés hiérarchiques des arbres dans certains de ses modèles (cf. chapitre 2 section 2.3.4).

2 Modèles d'illumination analytiques et hiérarchiques

2.1 Forme et illumination

Le détail d'un ensemble de primitives (*e.g.* une branche ou un arbre) n'est pas visible explicitement lorsque l'observateur est loin (*i.e.* lorsque l'ensemble des primitives ne recouvrent que quelques dizaines de pixels de l'image ou moins). Seule la forme et la couleur d'ensemble sont alors distinguées. L'idée que nous présentons ici est de représenter ce groupe de primitives par sa forme, associée à une formule analytique décrivant son comportement photométrique et son opacité globale (cf. figure 3.4), c'est-à-dire par un modèle d'illumination (*shader*). Ce *shader* est obtenu en intégrant un modèle d'illumination simple¹ (*e.g.* celui de Phong) sur l'ensemble des primitives du groupe en tenant compte de la visibilité², et en calculant leur opacité moyenne.

¹Les calculs mathématiques sur lesquels repose cette intégrale ne sont toutefois réalisables analytiquement que si la fonction d'illumination est simple.

²Ceci se traduit par une restriction du domaine d'intégration.

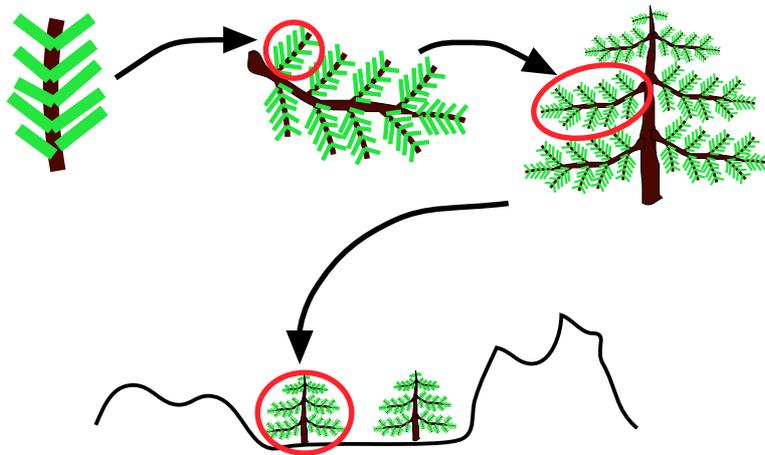


FIG. 3.3 – Les arbres sont hiérarchiques : plusieurs branches secondaires forment une branche principale, et plusieurs branches principales forment l'arbre. Toutes les branches se ressemblent, il est donc possible de les instancier. De même, un arbre peut être instancié plusieurs fois dans une scène. Il suffit alors de modifier son orientation et sa taille pour éviter l'impression d'uniformité.

Cette idée est présente dans beaucoup de modèles d'illumination de surface ou par exemple dans la représentation de fourrure de Kajiya puisqu'il intègre le modèle d'illumination de Phong sur un cylindre (cf. chapitre 2 section 1.2.2).

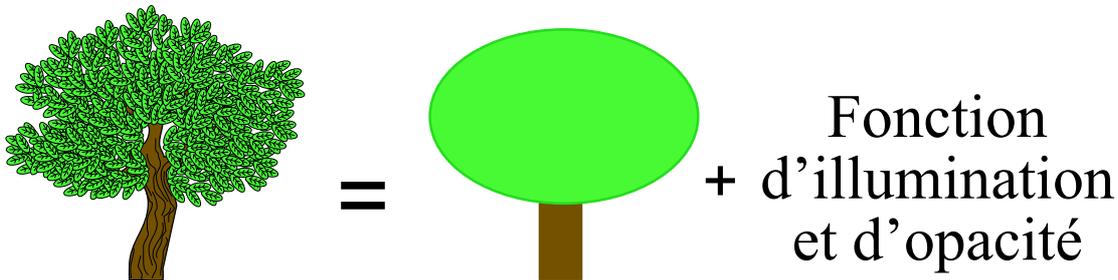


FIG. 3.4 – Un objet vu de loin peut être représenté par sa forme et son modèle d'illumination (*shader*). Le résultat de son rendu sera de meilleure qualité (*i.e.* avec peu d'aliasage), avec un temps de calcul plus petit que si on utilise la représentation où toute la géométrie est utilisée.

2.2 Hiérarchie

Le calcul de l'intégrale du modèle d'illumination sur les primitives n'est réalisable que si une connaissance a priori forte existe sur l'agencement des primitives. L'aspect hiérarchique des arbres dont nous parlions à la section 1, peut nous servir de base au calcul. Si nous essayons de calculer l'intégrale analytique de la fonction de Phong sur un arbre entier il est probable que le calcul ne soit pas réalisable (la géométrie trop complexe d'un arbre ne pourra se traduire par une modélisation mathématique raisonnable pour les calculs). En revanche, nous pouvons procéder par étapes, tout en construisant les niveaux de détails. Nous commençons par calculer le *shader* I_{branche} d'une branche en intégrant le modèle illumination de Phong sur l'ensemble de ses primitives (*e.g.* en fixant une géométrie simple des feuilles). En utilisant ce résultat, nous calculons le *shader* I_{arbre} d'un arbre entier, en calculant l'intégrale de I_{branche} sur l'ensemble des branches (cf. figure 3.5).

De plus, un terme d'opacité moyenne doit être calculé pour chaque niveau de la hiérarchie. La description que nous venons de faire se limite à deux niveaux dans un but pédagogique ; il est bien sûr préférable d'en avoir plus.

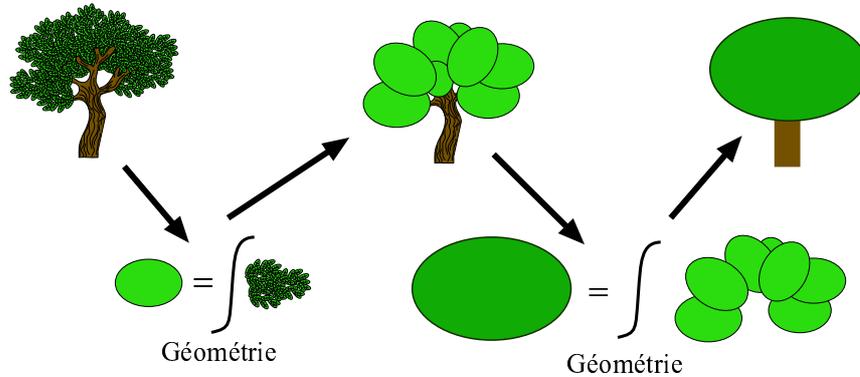


FIG. 3.5 – Principe d'une hiérarchie de *shaders* analytiques dans le cas d'un arbre. L'illumination analytique d'une branche est calculée en intégrant la fonction d'illumination de base (e.g. le modèle de Phong) sur l'ensemble des feuilles d'une branche (une connaissance a priori de la disposition des feuilles est nécessaire pour simplifier la modélisation mathématique). À partir de cette fonction d'illumination (et toujours de la connaissance a priori de la répartition de la matière) on calcule finalement l'illumination de l'arbre entier. De plus, un terme d'opacité moyenne doit être calculé pour chaque niveau de la hiérarchie.

Nous venons de présenté une idée générale pour la construction de niveaux de détails d'arbres fondée sur l'intégration analytique du modèle d'illumination et sur leur hiérarchie naturelle, il nous reste maintenant à mettre en pratique ce concept pour prouver sa pertinence, ce que nous ferons au chapitre suivant. Avant ça, je tiens à préciser quelques points en rapport avec l'algorithme de rendu que nous avons utilisé.

3 Considérations liées au rendu

Les niveaux de détails, même basés sur une illumination analytique, doivent être utilisés à bon escient lors de la phase de rendu. Nous ferons quelques précisions concernant le lancer de cônes en 3.1, et introduirons en 3.2 un critère permettant le choix du "bon" niveau de détails.

3.1 Lancer de cônes

L'ombrage jouant un rôle important dans le réalisme des images, nous avons choisi d'utiliser un rendu par lancer de rayons. L'état de l'art nous indique que l'algorithme de *beam-tracing* permet de diminuer l'aliassage plus efficacement que le sur-échantillonnage dans le cas d'objets à hautes fréquences, comme les arbres (cf. chapitre 1 section 2.1.2). Du fait de la complexité du calcul exact d'intersection de l'algorithme de *beam-tracing*, nous avons choisi d'utiliser une adaptation de cet algorithme, le *cone-tracing* : pour chaque objet intersecté par notre rayon conique nous calculons son pourcentage α de recouvrement de la section du rayon. Le calcul global de la couleur d'un pixel est :

$$C_{\text{pixel}} = a_A \cdot \alpha_A \cdot C_A + (1 - a_A \cdot \alpha_A) \times a_B \cdot \alpha_B \cdot C_B + \dots$$

avec a_A le pourcentage de recouvrement du pixel par l'objet A et α_A l'opacité de l'objet A, etc. Le calcul du pourcentage de recouvrement ne tient pas compte de la répartition géométrique

des objets dans le pixel. Néanmoins cette approximation est valide si les objets sont distribués uniformément dans l'espace, du moins sans corrélation (ce qui est le cas des arbres).

Ce schéma de calcul constitue un bon compromis temps de calcul/aliassage. Nous pouvons ainsi lancer un seul rayon par pixel, sans pour autant surcharger le calcul d'une précision superflue (pour les arbres) qu'apporterait le *beam-tracing*. Remarque : ce schéma de calcul est similaire à celui de l'algorithme du *A-buffer* (qu'il aurait été possible d'utiliser ici) qui calcule de la même manière un pourcentage de recouvrement du pixel.

3.2 Choix du niveau de détails

Durant la phase de rendu il faut choisir le niveau de détails à utiliser. Celui-ci doit avoir une résolution légèrement inférieure à la taille d'un pixel une fois qu'il est projeté à l'écran (*cf.* figure 3.6) :

- une résolution trop fine conserve les problèmes d'une représentation sans niveaux de détails, à savoir un traitement de la visibilité complexe et un nombre de primitives à traiter important : le coût reste élevé et l'aliassage toujours présent (*cf.* figure 3.6 à gauche).
- une résolution trop grossière est immédiatement visible par l'utilisateur car l'ensemble de l'objet est uniforme et les fréquences sont gommées (*cf.* figure 3.6 à droite).

Le choix du bon niveau de détails est donc fonction de la taille de la primitive à l'écran, qui elle dépend de la distance entre l'objet et l'observateur.

Remarque : en acceptant un léger compromis sur la qualité, l'utilisateur peut forcer l'utilisation d'un niveau de détail légèrement au delà de sa limite de validité, diminuant ainsi la charge de calcul. L'utilisation d'un niveau de détails grossier hors de sa limite de validité tend à donner des images floues, ce qui peut être un avantage pour les ombres, ainsi que pour simuler la profondeur de champ.

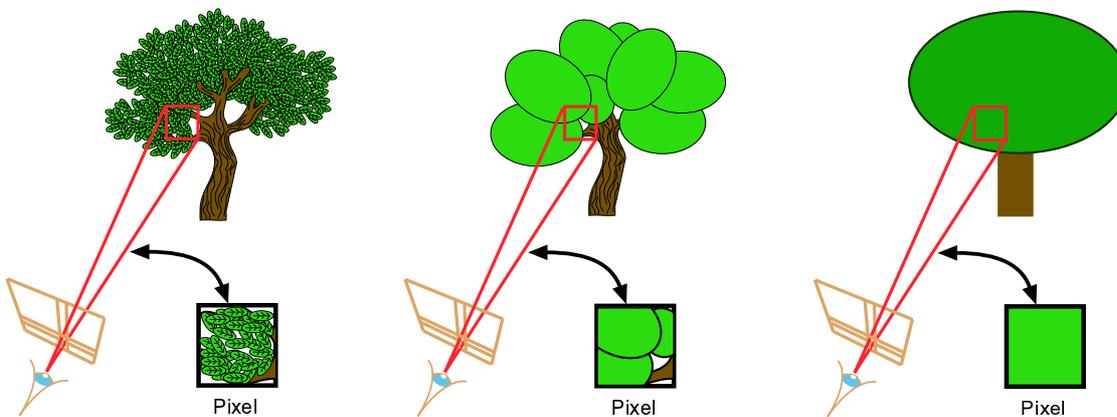


FIG. 3.6 – Le bon niveau de détails est choisi en fonction de l'éloignement de l'objet à l'œil mais aussi de la taille de ses primitives. À gauche : de nombreuses feuilles se trouvent dans le pixel ; on retombe dans les problèmes liés au grand nombre de détails (visibilité complexe, aliassage, coût élevé). Au milieu : une poignée de branches se trouvent dans le pixel ; le traitement sera rapide et ne fera pas apparaître de zone uniforme. Ce niveau de détails est le bon choix pour ce pixel. À droite : l'ellipse représentant l'arbre entier recouvre plusieurs pixels, ce qui donnera un aspect uniforme non réaliste à l'arbre.

Modèle analytique et hiérarchique d'illumination pour les conifères

La complexité d'un sapin est grande, plusieurs centaines de milliers d'aiguilles. Cependant, le détail de chaque aiguille n'est pas visible explicitement dès que l'observateur est éloigné de l'arbre. Dans ce chapitre nous appliquons l'idée de remplacer les données non visibles explicitement par une primitive "floue" reproduisant le même comportement photométrique que les géométries qu'elle représente. On dispose d'une connaissance a priori sur la géométrie d'un rameau de conifères, puisqu'une aiguille a une forme proche du cylindre et que leurs répartitions sur une branche est assez simple à caractériser (*cf.* Étude de cas).

Nous introduisons plusieurs primitives analytiques pour différents niveaux de détails. En 1 nous posons les hypothèses et présenterons les concepts de notre modèle, puis nous présenterons les trois niveaux du modèle : en 2 le *shader* représentant une aiguille, en 3 le *shader* représentant une révolution d'aiguilles (cône d'aiguilles) et en 4 le *shader* représentant un rameau complet. Enfin, nous finirons en 5 par les résultats et en 6 par la conclusion.

1 Shader dédiés aux conifères

Nous posons ici les hypothèses et les bases de notre modèle avec, en 1.1 notre modèle d'arbre, en 1.2 une description des trois modèles d'illumination ou *shaders* formant la hiérarchie, et en 1.3 le détail de ce que nous devons calculer.

1.1 Notre modèle d'arbres (*cf.* figure 4.1)

- Un arbre est un ensemble de branches et d'aiguilles que nous construisons en utilisant un *L-system*.

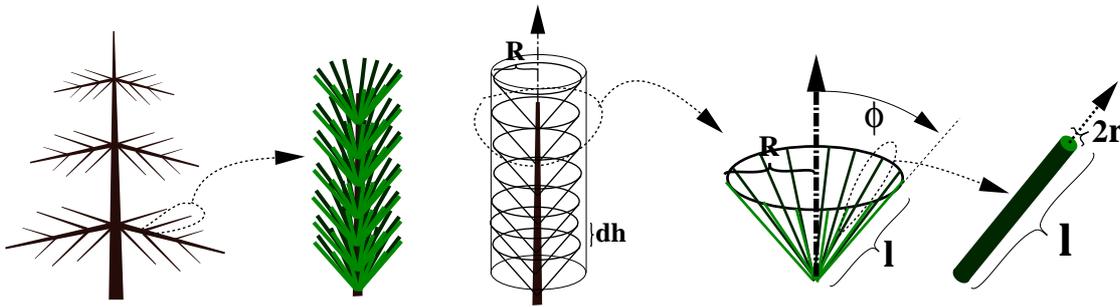


FIG. 4.1 – Notre description hiérarchique d'un arbre.

- Les branches sont représentées par de la géométrie classique.
- Les aiguilles sont des cylindres faisant un angle ϕ avec la branche, de longueur l , de rayon r et de densité (*i.e.* de distribution) ρ . Les paramètres changent peu le long de la branche, on peut les considérer comme constants.
- La projection d'une aiguille sur le plan perpendiculaire à la branche est de $R = l \cdot \sin(\phi)$.
- Nous supposons que les aiguilles sont distribuées en cônes, avec N aiguilles par cône. La distance entre deux cônes le long de la branche est dh . L'espace entre deux aiguilles est $\frac{2\pi R}{N}$, comme l'espace entre deux cônes est dh , il est raisonnable de choisir $dh = \frac{2\pi R}{N} = \sqrt{\rho}$. Nous avons donc la relation $dh \frac{2\pi R}{N} = \rho$.

1.2 Rendu multi-échelle

En fonction de la distance, la plus petite primitive que nous utilisons à l'affichage est l'aiguille (niveau un), le cône (niveau deux) ou la branche (niveau trois). Nous rendons la scène en utilisant un lancer de cônes : le rayon conique est utilisé pour estimer la taille apparente de la primitive et pour calculer le recouvrement α du pixel. Nous utilisons aussi des rayons coniques pour l'ombrage, en supposant que la source de lumière est ponctuelle.

Le principal problème est de calculer la réflectance globale et l'opacité des primitives considérées en incluant les ombres internes. Puisque nous utilisons des rayons coniques, le sur-échantillonnage¹ est inutile (*i.e.* un unique rayon par pixel est lancé).

La contribution essentielle que nous apportons est la représentation multi-échelles, qui sera détaillée dans les sections suivantes, les trois *shaders* alors obtenus par intégration et la méthode que nous utilisons pour résoudre ces intégrales, en particulier l'interprétation géométrique de la visibilité et de l'ombrage effectuée pour le calcul du niveau 3.

1.3 Qu'avons-nous à calculer ?

Dans cette section nous estimons le travail à effectuer pour le calcul analytique des *shaders*. Le résultat et le détail des intégrations successives se trouvent dans les trois prochaines sections. Les vecteurs \vec{L} et \vec{V} sont considérés comme constants pour l'objet car la source de lumière et le point de vue sont éloignés.

¹Le sur-échantillonnage peut être vu comme une approximation numérique du lancer de cônes (*cf.* chapitre 1 section 2.1.2).

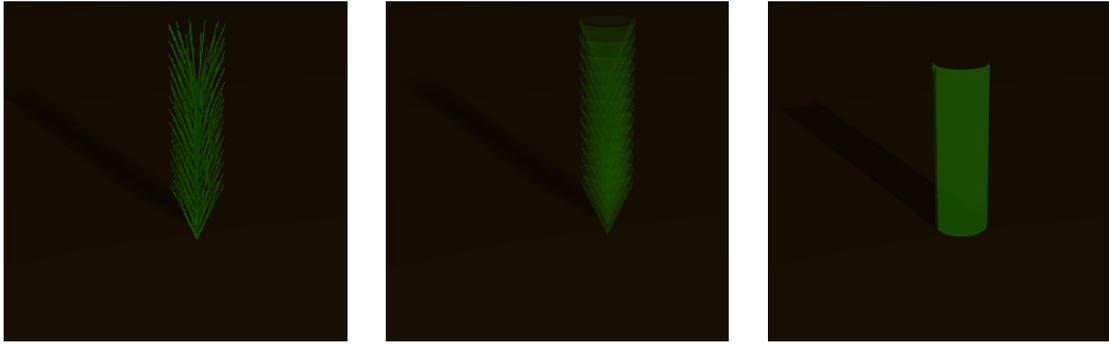


FIG. 4.2 – Les trois niveaux de modèles d’illumination ou *shaders*. À *droite* : la primitive est le cylindre représentant les aiguilles. Au *milieu* : la primitive est le cône représentant une révolution d’aiguilles. À *gauche* : la primitive est le cylindre représentant un rameau d’aiguilles.

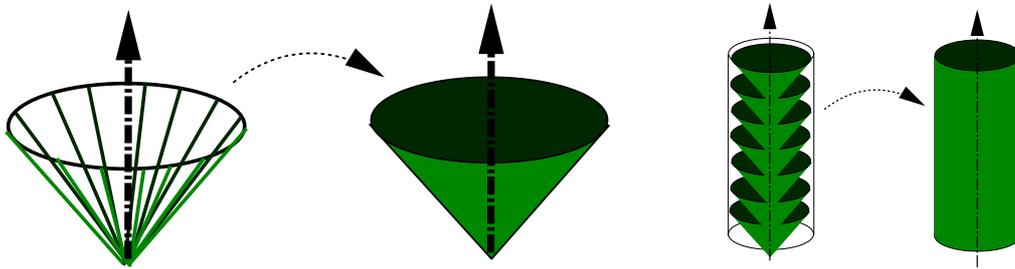


FIG. 4.3 – À *gauche* : le modèle de cône continu. À *droite* : le modèle de rameau continu.

1.3.1 Niveau un (aiguilles)

Pour calculer l’illumination d’une aiguille, nous devons intégrer la lumière diffuse I_d et spéculaire I_s réfléchées par un cylindre (cf. chapitre 2 section 1.2.2). Nous ne calculons jamais explicitement l’intersection entre une aiguille et le rayon conique, mais l’intersection du rayon avec le cône d’aiguilles. Nous considérons que les aiguilles visibles sont sur la partie avant du cône, puis nous sommes les illuminations.

1.3.2 Niveau 2 (cônes)

En appliquant la loi des grands nombres, nous considérons que l’illumination d’un cône d’aiguilles est équivalente à l’illumination d’un cône semi-opaque continu où chaque point reflète la lumière comme une aiguille entière le ferait (cf. figure 4.3). L’opacité A est la moyenne de la surface du cône couverte par les aiguilles, ce qui donne $A = \frac{2N_f}{\pi R}$. L’illumination totale réfléchie est A multiplié par l’intégrale dans l’espace des pixels de l’illumination d’un cylindre sur la partie visible du cône. Les parties avant et arrière du cône sont considérées séparément, et seule une portion de cette partie peut être visible dans un pixel. L’intégration analytique n’est pas triviale, et nécessite certaines approximations.

1.3.3 Niveau 3 (rameaux)

Nous considérons que le *shader* d’un rameau d’aiguilles est équivalent à un cylindre volumique, anisotrope et semi-opaque, fait d’une imbrication de cônes (cf. figure 4.3). L’illumination et l’opacité de la partie avant et arrière du cône correspondent à celles du deuxième niveau que

nous avons vu précédemment. En appliquant la loi des grands nombres, le volume est considéré comme continu et anisotrope : l'opacité doit reproduire le même effet que celui produit par tous les cônes traversés par un rayon si le rendu se faisait avec le niveau 2, ce qui dépend fortement de l'angle du rayon (cf. figure 4.10). La partie difficile est l'intégration volumique analytique, en tenant compte de la visibilité et de l'ombrage. En supposant que l'on puisse utiliser une approximation linéaire de la loi de composition des opacités, *i.e.* $(1 - A)^n \approx (1 - nA)$ valide aux opacités faibles, nous transformerons cette intégrale en une forme géométrique.

2 Shader analytique d'une aiguille

Dans notre modèle, une aiguille est représentée par un cylindre (cf. chapitre 2 section 1.2.2), ayant comme *shader* celui de Phong. Nous devons donc intégrer les composantes diffuses et spéculaires d'un cylindre dans l'espace écran (*i.e.* nous devons sommer cette contribution dans le pixel en tenant compte de l'opacité).

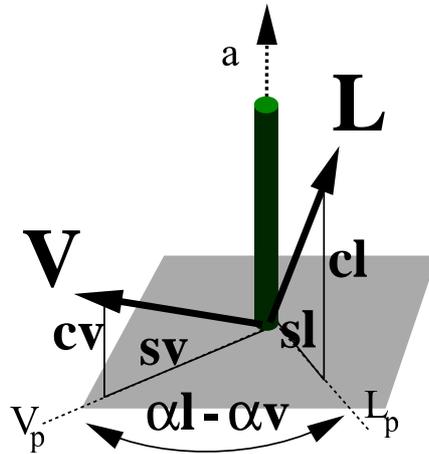


FIG. 4.4 – Une aiguille.

2.1 Illumination diffuse

La composante diffuse depuis le point de vue est

$$\begin{aligned} R_d^{cyl} &= \frac{\int_{\text{pixel}} (N \cdot L) \mathbb{I}_{(N \cdot L > 0)} dS_{\text{pix}}}{\int_{\text{pixel}} 1 dS_{\text{pix}}} \\ &= \frac{\int_{\text{cylindre}} (N \cdot L) \mathbb{I}_{(N \cdot L > 0)} \cdot (N \cdot V) \mathbb{I}_{(N \cdot V > 0)} dS}{\int_{\text{cylindre}} (N \cdot V) \mathbb{I}_{(N \cdot V > 0)} dS} \end{aligned}$$

Posons c_v et c_l les projections de V et L sur l'axe du cylindre \vec{a} , *i.e.* $c_v = (\vec{a} \cdot V)$ et $c_l = (\vec{a} \cdot L)$ (cf. figure 4.4). Posons V_p et L_p les projections de V et L sur le plan perpendiculaire au cylindre et s_v et s_l leur norme. Ce qui nous donne :

$$R_d^{cyl} = \frac{\int_{\alpha=\alpha_0}^{\alpha_1} s_l \cos(\alpha - \alpha_l) s_v \cos(\alpha - \alpha_v) d\alpha}{\int_{\alpha=\alpha_v - \frac{\pi}{2}}^{\alpha_v + \frac{\pi}{2}} s_v \cos(\alpha - \alpha_v) d\alpha}$$

avec α_V et α_L l'angle entre V_p et L_p dans le plan. Les bornes de visibilité α_0 et α_1 sont $\alpha_V - \frac{\pi}{2}$ et $\alpha_L + \frac{\pi}{2}$ si $L \times V$ a la même direction que \vec{a} , sinon ces bornes sont $\alpha_0 = \alpha_L - \frac{\pi}{2}$ et $\alpha_1 = \alpha_V + \frac{\pi}{2}$. Nous introduisons $\Delta\alpha = |\alpha_V - \alpha_L|$, et obtenons alors :

$$\begin{aligned} R_d^{cyl} &= \frac{s_l s_v / 2 (\sin(\Delta\alpha) + (\pi - \Delta\alpha) \cos(\Delta\alpha))}{2s_v} \\ \boxed{R_d^{cyl} &= \frac{s_l}{4} (\sin(\Delta\alpha) + (\pi - \Delta\alpha) \cos(\Delta\alpha))} \end{aligned} \quad (4.1)$$

2.2 Illumination spéculaire

L'illumination spéculaire depuis le point de vue est :

$$\begin{aligned} R_s^{cyl} &= \frac{\int_{\text{pixel}} (\mathbf{N} \cdot \mathbf{H})^n \mathbf{I}_{(\mathbf{N} \cdot \mathbf{H} > 0)} \cdot dS_{\text{pix}}}{\int_{\text{pixel}} 1 \cdot dS_{\text{pix}}} \\ &= \frac{\int_{\text{cylindre}} (\mathbf{N} \cdot \mathbf{H})^n \mathbf{I}_{(\mathbf{N} \cdot \mathbf{H} > 0)} \cdot (\mathbf{N} \cdot \mathbf{V}) \mathbf{I}_{(\mathbf{N} \cdot \mathbf{V} > 0)} \cdot dS}{\int_{\text{cylindre}} (\mathbf{N} \cdot \mathbf{V}) \mathbf{I}_{(\mathbf{N} \cdot \mathbf{V} > 0)} \cdot dS} \end{aligned}$$

avec le demi-vecteur $\mathbf{H} = \frac{\mathbf{V} + \mathbf{L}}{|\mathbf{V} + \mathbf{L}|}$ et n l'exposant spéculaire. Posons H_p , c_h , s_h et α_V défini comme pour L et V . Alors :

$$R_s^{cyl} = \frac{\int_{\alpha=\alpha_0}^{\alpha_1} s_h^n \cos^n(\alpha - \alpha_H) s_v \cos(\alpha - \alpha_V) d\alpha}{\int_{\alpha=\alpha_V - \frac{\pi}{2}}^{\alpha_V + \frac{\pi}{2}} s_v \cos(\alpha - \alpha_V) d\alpha}$$

Il est connu que $\cos^n(x)$ est similaire à $e^{-\frac{n}{2}x^2}$ pour n grand (ce qui est la cas ici). De plus, la densité de la fonction est concentrée sur $x = 0$ (l'écart type vaut $1/\sqrt{n}$, et n est généralement plus grand que 100), alors nous avons $\cos^n(x - x_0)f(x) \approx \cos^n(x - x_0)f(x_0)$. Finalement nous obtenons :

$$R_s^{cyl} \approx (s_h^n s_v \cos(\alpha_H - \alpha_V) \int_{\alpha=\alpha_0}^{\alpha_1} e^{-\frac{n}{2}(\alpha - \alpha_H)^2} d\alpha) / 2s_v$$

Puisque $\int_{-\infty}^{\infty} e^{-\frac{1}{2}(\frac{x}{\sigma})^2} = \sqrt{2\pi}\sigma$, l'intégrale précédente devient $\sqrt{\frac{2\pi}{n}}$ si $\alpha_H \in [\alpha_0, \alpha_1]$ (ce qui est toujours le cas). Donc :

$$\boxed{R_s^{cyl} \approx \frac{1}{2} s_h^n \cos(\alpha_H - \alpha_V) \sqrt{\frac{2\pi}{n}}} \quad (4.2)$$

2.3 Opacité

L'opacité est la proportion du rectangle de l'aiguille apparent qui se projette dans le pixel. Si l'aiguille est totalement couverte par le pixel alors :

$$\boxed{\text{alpha}^{cyl} = \frac{2r \cdot s_v l}{S_{\text{pix}}}} \quad (4.3)$$

où S_{pix} représente la surface de la section du rayon conique à la distance de la primitive. L'illumination diffuse et spéculaire valent donc $I_d = \text{alpha} R_d$ et $I_s = \text{alpha} R_s$.

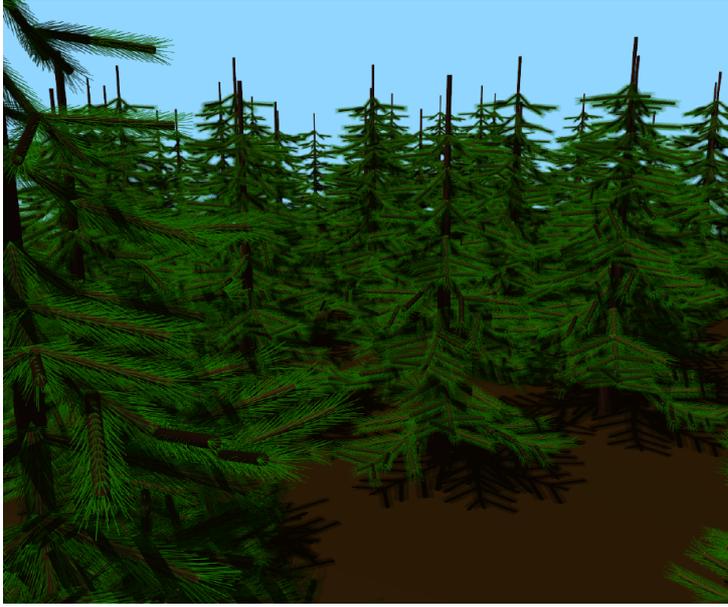


FIG. 4.5 – Notre *shader* de cylindre permet de représenter une forêt de sapins dont les aiguilles sont considérées comme cylindriques sans avoir à les trianguler ou à les échantillonner.

3 Shader analytique d'un cône d'aiguilles

Comme nous l'avons vu à la section 1.3.2, nous considérons à cette échelle qu'un cône d'aiguilles est une surface continue semi-opaque d'opacité A où chaque point de la surface a une réflexion identique à celle d'un cylindre. Nous devons donc intégrer l'illumination d'un cylindre sur un cône d'ouverture ϕ pour tous les axes \vec{a}_θ d'aiguilles valides. Dans le système de coordonnées polaires associé au cône nous avons $L = (\theta_L, \phi_L)$, où ϕ_L est l'angle entre L et l'axe du cône. De même nous avons $V = (\theta_V, \phi_V)$.

3.1 Illumination diffuse

L'illumination diffuse est donnée par :

$$\begin{aligned} I_d^{\text{cône}} &= A \int_{\theta=\theta_V-\frac{\pi}{2}}^{\theta_V+\frac{\pi}{2}} l_s s_v R_d^{\text{cyl}}(\vec{a}_\theta) \\ &= \frac{l_s A}{4} \int_{\theta=\theta_V-\frac{\pi}{2}}^{\theta_V+\frac{\pi}{2}} s_l s_v (\sin(\Delta\alpha) + (\pi - \Delta\alpha) \cos(\Delta\alpha)) \end{aligned}$$

où l_s est la longueur apparente d'une aiguille. Cette formule n'est pas intégrable analytiquement. Nous approchons alors $s_l s_v (\sin(\Delta\alpha) + (\pi - \Delta\alpha) \cos(\Delta\alpha))$ en utilisant la fonction²

$$F = s_l s_v \cdot \left(\frac{1 + \cos(\Delta\alpha)}{2} \right) (2 + (\pi - 2) \cos(\Delta\alpha))$$

qui a la même valeur et la même dérivée en 0 , $\frac{\pi}{2}$ et π . De plus, l'erreur maximale entre les deux fonctions est plus petite que 1%³.

²Nous avons trouvé les différentes formules approximantes par tâtonnement, en cherchant des formes intégrables et en contrôlant les différences sous *Maple*.

³L'approximation de F par $\pi \left(\frac{1 + \cos(\Delta\alpha)}{2} \right)^{1.6515}$ est aussi très bonne : l'erreur est plus petite que 1.5%. Il est plus

Si

$$\cos(\Delta\alpha) = \frac{(L_p \cdot V_p)}{(|L_p| \cdot |V_p|)} = \frac{(L \cdot V) - c_l c_v}{s_l s_v}$$

alors

$$\int F = (L \cdot V + s_l s_v - c_l c_v) \cdot (2 + (\pi - 2)(L \cdot V - c_l c_v) / s_l s_v)$$

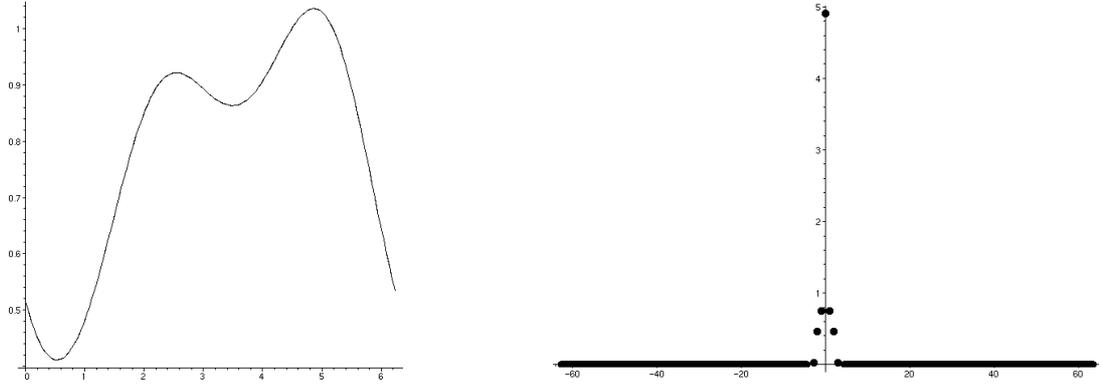


FIG. 4.6 – À gauche un exemple de la courbe F , pour $L = (0, 1.2)$, $V = (1, 1.5)$ et $\phi = 0.5$. Cette courbe est plutôt lisse malgré l'aspect complexe de ses facteurs. À droite : la FFT de cette courbe. Notez que l'énergie est clairement concentrée dans les fréquences 0, 1 et 2, la motivation d'approcher F avec une combinaison linéaire de 1 , $\cos(\theta - \theta_A)$, $\cos(2(\theta - \theta_B))$ est donc justifiée. NB : les valeurs à droite sont dues à la symétrie de la FFT.

Si nous traçons cette fonction avec *Maple* pour plusieurs valeurs des paramètres L , V et ϕ , il apparaît que la courbe est très lisse (cf. figure 4.6 à gauche) et ressemble à une combinaison linéaire de 1 , $\cos(\theta - \theta_A)$ et $\cos(2(\theta - \theta_B))$. L'évaluation de la FFT sur des courbes discrétisées montre qu'il n'y a pratiquement pas d'énergie en dehors des fréquences 0, 1 et 2 (cf. figure 4.6 à droite). Nous essayons donc d'approcher la courbe F en utilisant des positions et des valeurs des extrema. Le premier facteur est responsable d'un maximum de variation de F et est facile à calculer. De ce fait, nous approchons la courbe F par $(L \cdot V) + s_l s_v - c_l c_v$ dont les extrema correspondent aux mêmes valeurs de θ que F .

Le terme $c_l c_v - s_l s_v$ est égal à $\cos(\widehat{AL} + \widehat{AV})$ avec \widehat{AL} l'angle entre les vecteurs \vec{a} et L , et \widehat{AV} l'angle entre les vecteurs \vec{a} et V . Ces angles varient peu entre leur minimum et leur maximum si le vecteur \vec{a} tourne autour du cône, nous représentons donc les variations de \widehat{AL} par la forme $A_L + B_L \cos(\theta - \theta_L)$ avec $A_L = \max(\phi_L, \phi)$, $B_L = \min(\phi_L, \phi)$. Nous faisons de même pour \widehat{AV} .

Si nous développons $\widehat{AL} + \widehat{AV}$ avec cette approximation nous obtenons l'expression :

$$A_\Sigma + B_\Sigma \cos(\theta - \theta_\Sigma)$$

avec

$$\begin{aligned} A_\Sigma &= A_L + A_V \\ B_\Sigma^2 &= B_L^2 + B_V^2 + 2B_L B_V \cos(\theta_L - \theta_V) \\ \cos(\theta_\Sigma) &= (B_L \cos(\theta_L) + B_V \cos(\theta_V)) / B_\Sigma \\ \sin(\theta_\Sigma) &= (B_L \sin(\theta_L) + B_V \sin(\theta_V)) / B_\Sigma \end{aligned}$$

simple d'utiliser une puissance de 1.5, 2 ou 1 (avec des erreurs respectives de 4%, 7%, 18% à la place de $\frac{\log(\pi)}{\log(2)}$), pour résoudre l'intégrale analytiquement.

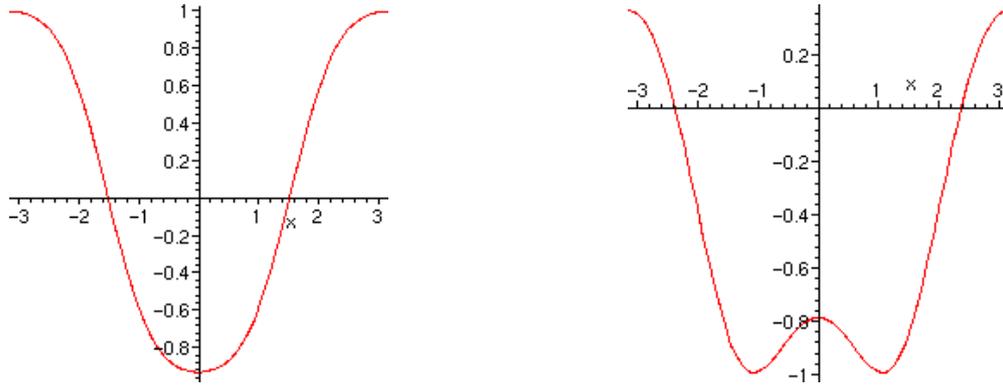


FIG. 4.7 – Les deux types d'aspects de la courbe $\cos(A_\Sigma + B_\Sigma * \cos(\theta - \theta_\Sigma))$, dépendant de ce que $A_\Sigma + B_\Sigma * \cos(\theta - \theta_\Sigma)$ croise π (à droite) ou non (à gauche).

Notre but étant d'approcher F , nous cherchons les extrema de $F \approx (L.V) - \cos(\widehat{AL} + \widehat{AV})$. Ils correspondent aux extrema de $\widehat{AL} + \widehat{AV}$, soit à la valeur pour laquelle $\widehat{AL} + \widehat{AV}$ croise π . Si $\widehat{AL} + \widehat{AV}$ ne croise pas π , F est similaire à une fonction cosinus. Si elle le croise, F a un "chapeau" et ressemble à la combinaison d'un cosinus et d'un cosinus à fréquence double (cf. figure 4.7). La similitude est grande si ϕ_L et ϕ_V ne sont pas proches de ϕ . Alors, nous pouvons obtenir explicitement les extrema de la courbe.

Comme nous essayons précisément d'approcher F sous la forme

$$(L.V) - (\lambda_0 + \lambda_1 \cos(\theta - \theta_m) + \lambda_2 \cos(2(\theta - \theta_m)))$$

nous pouvons utiliser les paramètres issus de ces extremum.

Soit $M = \cos(A_\Sigma - B_\Sigma)$ et $m = \cos(A_\Sigma + B_\Sigma)$, alors

$$\theta_m = \theta_\Sigma,$$

$$\lambda_1 = \frac{(m-M)}{2},$$

$\lambda_0 = \frac{(m+M)}{2} - \lambda_2$, avec $\lambda_2 = 0$ dans le cas où aucun croisement avec π n'apparaît (i.e. $A_\Sigma + B_\Sigma$ et $A_\Sigma - B_\Sigma$ sont entre $[0, \pi]$);

$\lambda_2 = \frac{\lambda_1 B_\Sigma}{4(2\pi - A_\Sigma)}$ dans le cas d'un croisement avec π ($A_\Sigma + B_\Sigma > \pi > A_\Sigma - B_\Sigma$).

Maintenant, nous pouvons facilement obtenir l'intégrale de F :

$$I_d^{\text{cone}} = \frac{lA}{4} \int_{\theta=\theta_V-\frac{\pi}{2}}^{\theta_V+\frac{\pi}{2}} F$$

$$\boxed{I_d^{\text{cone}} = \frac{lA}{4} (\pi(LV - \lambda_0) - 2\lambda_1 \cos(\theta_V - \theta_\Sigma))} \quad (4.4)$$

où

$$\cos(\theta_V - \theta_\Sigma) = \frac{B_L \cos(\Delta\theta) + B_V}{\sqrt{B_L^2 + B_V^2 + 2B_L B_V \cos(\Delta\theta)}} \text{ et } \Delta\theta = \theta_L - \theta_V.$$

3.2 Illumination spéculaire

L'illumination spéculaire est donnée par :

$$I_s^{\text{cone}} = A \int_{\theta=\theta_V-\frac{\pi}{2}}^{\theta_V+\frac{\pi}{2}} l.s_v R_s^{\text{cyl}}(\vec{a}_\theta)$$

$$= \frac{l.A}{2} \sqrt{\frac{2\pi}{n}} \int_{\theta=\theta_V-\frac{\pi}{2}}^{\theta_V+\frac{\pi}{2}} s_l s_h^n \cos(\alpha_H - \alpha_V)$$

avec $l s_v$ la longueur apparente d'une aiguille. Encore une fois, s_h^n est une fonction dont la densité est concentrée quand $s_h = 1$, ce qui arrive quand $c_h = 0$, c'est-à-dire quand H est orthogonal à l'axe d'une aiguille \vec{a} . Donc la valeur θ_H^\perp existe seulement si $\phi_H \in [\frac{\pi}{2} - \phi, \frac{\pi}{2} + \phi]$, sinon $I_s^{\text{cone}} = 0$. Si θ_H^\perp existe, nous avons à nouveau $s_h^n f(\theta) \approx s_h^n f(\theta_H^\perp)$.

Comme $s_l s_h \cos(\alpha_H - \alpha_V) = (V.H) - c_h c_v$, nous obtenons finalement :

$$\boxed{I_s^{\text{cone}} \approx \frac{l.A}{2} \frac{2\pi}{n} (V.H) \epsilon} \quad (4.5)$$

où $\epsilon = 1$ si $\phi_H \in [\frac{\pi}{2} - \phi, \frac{\pi}{2} + \phi]$, sinon $\epsilon = 0$.

Remarque : quand les deux valeurs où H est orthogonal à \vec{a} apparaissent sur la même face (avant ou arrière), $\epsilon = 2$.

3.3 Opacité

L'opacité est donnée par :

$$\text{alpha}^{\text{cone}} = A \int_{\theta=\theta_V-\frac{\pi}{2}}^{\theta_V+\frac{\pi}{2}} l s_v$$

Comme $s_v = \sin(\widehat{AV})$, nous approchons \widehat{AV} by $A_V + B_V \cos(\theta - \theta_V)$ par le même moyen que pour la composante diffuse. C'est-à-dire :

$$\text{alpha}^{\text{cone}} = lA(\lambda_0^V \pi + 2\lambda_1^l \cos(\theta_V - \theta_V)) , \text{ i.e}$$

$$\text{alpha}^{\text{cone}} = l.A(\pi \cos(\phi) \cos(\phi_V) - 2 \sin(\phi) \sin(\phi_V)) \quad (4.6)$$

4 Modèle d'illumination analytique d'un rameau d'aiguilles

Nous considérons qu'un rameau d'aiguilles est un volume ayant une forme de cylindre et une opacité anisotrope (cf. figure 4.9). Pour calculer le *shader* analytique d'un tel objet nous devons calculer analytiquement un rendu volumique du cylindre.

Comme l'opacité A n'est pas constante le long du rayon et du rayon d'ombrage, nous obtenons :

$$I = \frac{1}{S_{\text{pix}}} \int_{(x,y) \in \text{pixel}} \int_{z=\text{near}}^{\text{far}} A I^{\text{cyl}} e^{-\int_0^{l_z} \sigma} e^{-\int_0^{l_{\text{shad}}} \sigma} \quad (4.7)$$

avec $e^{-\sigma} = T = (1 - A)$ la transparence anisotrope, l_z la longueur du rayon dans le volume et l_{shad} la longueur du rayon d'ombrage dans le volume.

Nous devons maintenant exprimer l'opacité et calculer l'intégrale, ce qui requière quelques approximations.

4.1 Traversée d'un rameau d'aiguilles en 2D

Nous nous plaçons dans le cas d'une branche infinie avec un empilement d'aiguilles ayant une direction ϕ relativement à l'axe de la branche (*cf.* figure 4.8 à gauche). Posons R le rayon de la branche et dh la distance verticale entre deux cônes d'aiguilles. Soit un rayon traversant la branche, faisant un angle ϕ_r avec l'axe de celle-ci. La longueur du rayon à l'intérieur de la branche est $R/\sin(\phi_r)$.

Les distances entre les intersections sont $\delta = dh \frac{\sin(\phi)}{\sin|\phi_r - \phi|}$

Le nombre moyen d'intersections est $\frac{R}{dh} \frac{\sin|\phi_r - \phi|}{\sin(\phi)\sin(\phi_r)}$

Notons $k(\phi_r, \phi)$ la quantité

$$k(\phi_r, \phi) = \frac{\sin|\phi_r - \phi|}{\sin(\phi)\sin(\phi_r)} = \left| \frac{1}{\tan(\phi)} - \frac{1}{\tan(\phi_r)} \right|$$

L'opacité de la branche le long du rayon est $1 - T^{\frac{R}{dh}k(\phi_r, \phi)}$

Afin de raccourcir les notations, notons $k_r = k(\phi_r, \phi)$ et $\bar{k}_r = k(\phi_r, \pi - \phi)$. \bar{k}_r correspond à la traversée d'une branche ayant une symétrie axiale.

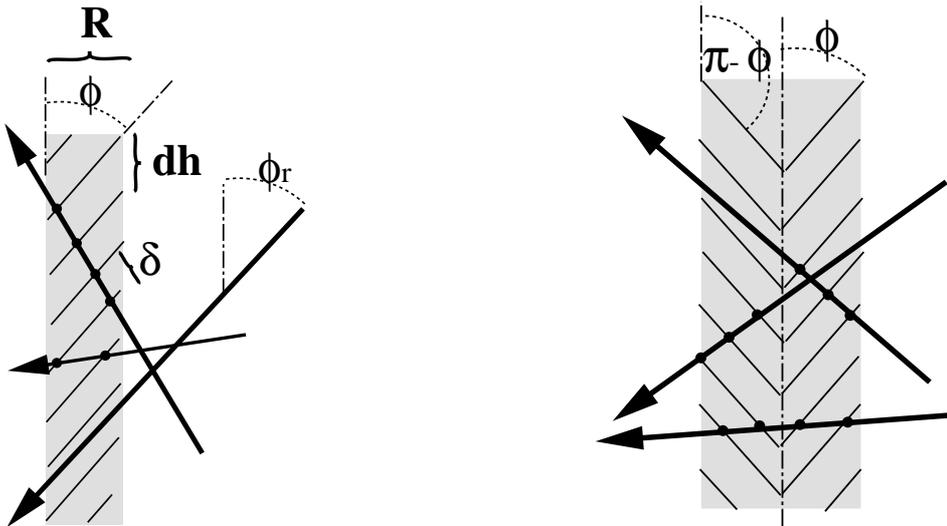


FIG. 4.8 – À gauche : champ 2D d'aiguilles parallèles. À droite : rameau d'aiguilles 2D. Notez la variation de l'opacité en fonction de la direction du rayon (surtout visible à gauche).

Un rameau d'aiguilles 2D est composé de deux champs d'aiguilles différents, celui de droite ayant pour orientation d'aiguilles ϕ et celui de gauche $\pi - \phi$ (*cf.* figure 4.8 à gauche). Le nombre total d'intersections le long du rayon est donc :

$$\frac{R}{dh}(k_r + \bar{k}_r) = \frac{R}{dh} \frac{\sin|\phi_r - \phi| + \sin|\phi_r + \phi|}{\sin(\phi)\sin(\phi_r)} = \frac{R}{dh} \frac{2}{\tan(\min(\phi, \phi_r))}$$

Ce qui signifie que pour un rayon restant dans l'ouverture du cône d'aiguilles, l'opacité totale reste constant et ce, quelle que soit la partie du cône qui est devant et qui est derrière. Ceci est également vrai pour le rayon d'ombre. Si le rayon passe par l'ouverture du cône (par en haut ou par en bas) l'opacité passe à 100% quand $\phi_r = 0$ ou π .

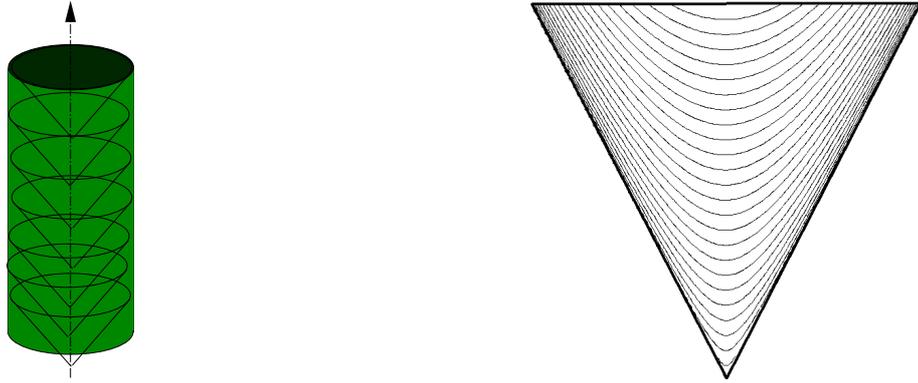


FIG. 4.9 – À gauche : nous modélisons un rameau d'aiguilles par un cylindre volumétrique semi-opaque. Cette opacité est anisotrope et reproduit la variation du nombre d'intersections entre le rayon et les cônes sous-jacents. À droite : intersection d'un plan P_x avec un cône. Nous approchons les hyperboles par leur asymptote.

4.2 Extension à la 3D

Revenons à notre rameau d'aiguilles en 3D. En 3D, si un rayon traverse l'axe de la branche, la situation est équivalente au cas 2D vu précédemment. Mais généralement, un rayon ne traverse pas exactement l'axe. Plaçons nous dans le plan parallèle à l'axe du cône contenant le rayon. Soit x la distance à l'axe et P_x ce plan. L'intersection du volume du rameau d'aiguilles (constitué de cônes) avec le plan donne un ensemble d'hyperboles. Nous approchons raisonnablement ces hyperboles par leurs deux asymptotes (*cf.* figure 4.9 à droite). Le plan contient les aiguilles ayant la même orientation ϕ et le même décalage dh qu'en 3D, dans une branche d'épaisseur $2R_x$ avec $R_x = \sqrt{R^2 - x^2}$. Nous pouvons donc calculer le nombre d'intersections en utilisant la formule 2D. Pour estimer la quantité de lumière atteignant un point du rayon, nous considérons le rayon d'ombre partant de ce point. De manière similaire, nous introduisons le plan parallèle à l'axe du cône et contenant ce rayon d'ombrage (*cf.* figure 4.10). Le nombre d'intersections peut être obtenu comme pour le rayon principal.

4.3 Traversée d'un rameau d'aiguilles 3D

Nous pouvons maintenant revenir à l'intégrale volumique (4.7). Nous choisissons la paramétrisation cartésienne de la surface de manière à ce que l'axe \vec{x} soit orthogonal au cylindre. Donc le plan P_x est indexé par x (*i.e.* x est cohérent avec la section précédente). En conséquence, il n'est pas nécessaire d'intégrer le long de l'axe \vec{y} , puisque le cylindre est homogène dans cette direction. Remarquez que l'albédo A de l'équation doit être corrigée en A/δ , puisque aucune énergie n'est présente dans l'espace entre deux cônes. De manière similaire pour la longueur dl , l'opacité est $e^{-\sigma dl} = T^{dl/\delta}$. Effectuons le changement de variable de (x, z) vers (x, z') dans le plan orthogonal au cylindre. Cela signifie que nous indexons un point sur le rayon par sa projection sur le plan orthogonal. Le jacobien de la transformation est $\frac{1}{\sin(\phi)}$. L'opacité associée à un élément de longueur dl' sur le plan est $T^{\frac{dl'}{\sin(\phi)\delta}} = T^{dl' \frac{k(\phi)}{dh}}$.

4.4 Division de l'intégrale en régions

Nous tirons comme information du cas 2D que l'opacité le long du rayon est constante pour la partie avant et pour la partie arrière (ces deux parties correspondent aux deux orientations des

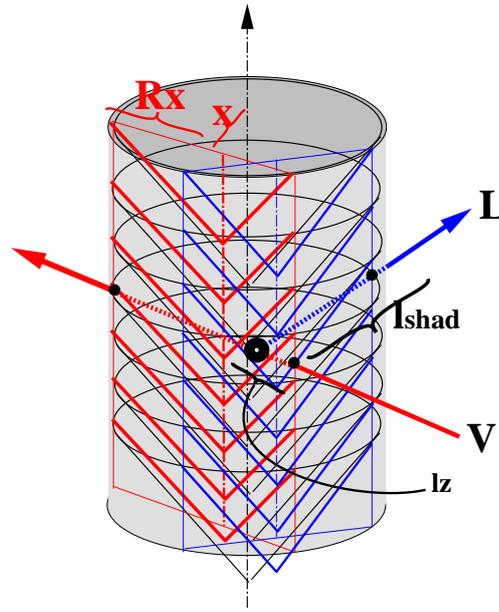


FIG. 4.10 – Le volume intersecté par le plan vertical contenant le rayon ressemble au cas 2D.

aiguilles dans le plan P_x). Nous avons donc :

$$I = \frac{1}{2R} \int_{x=-R}^R \left(\int_{z=-R_x}^0 A \frac{k_v}{dh} \cdot I^{cyl} \cdot T^{\frac{k_v}{dh}(R_x+z)} \cdot T^{\int_0^{l_{shad}} \frac{k(l)}{dh}} + \int_{z=0}^{R_x} A \frac{\bar{k}_v}{dh} \cdot I^{cyl} \cdot T^{\left(\frac{k_v}{dh}R_x + \frac{\bar{k}_v}{dh}z\right)} \cdot T^{\int_0^{l_{shad}} \frac{k(l)}{dh}} \right)$$

Nous divisons le disque (de la section du rameau d'aiguilles) en deux régions F_V et R_V (avant et arrière) relativement à V (cf. figure 4.11), tout comme à la section 3, nous avons divisé les cônes en une face avant et une face arrière, afin d'évaluer l'illumination. Sur chaque région $k()$ est constante. Supposons que I^{cyl} soit constante pour chacune des deux régions du volume, et approchons la par la valeur I_{front}^{cyl} et I_{rear}^{cyl} . L'intégrale devient

$$I = \frac{A}{2Rdh} \int_{x=-R}^R \left(k_v I_{front}^{cyl} \int_{z=-R_x}^0 T^{\frac{k_v}{dh}(R_x+z)} T^{\int_0^{l_{shad}} \frac{k(l)}{dh}} + \bar{k}_v I_{rear}^{cyl} \int_{z=0}^{R_x} T^{\left(\frac{k_v}{dh}R_x + \frac{\bar{k}_v}{dh}z\right)} T^{\int_0^{l_{shad}} \frac{k(l)}{dh}} \right)$$

Pour résoudre l'intégrale, nous allons diviser à nouveau le disque dans le but de séparer la partie avant et arrière F_L et R_L relativement à L . La longueur du rayon d'ombre dépend de z d'une manière complexe, ce qui rend l'exponentiel difficile à intégrer analytiquement.

Pour rendre cette intégrale réalisable analytiquement, nous utilisons une approximation linéaire de composition des opacités : $(1 - A)^n \approx (1 - nA)$ qui est valide si $nA \ll 1$ (i.e. si le rameau d'aiguilles n'est pas trop dense). Alors $(1 - A)^{n_1}(1 - A)^{n_2} \approx 1 - n_1A - n_2A$, ce qui assure la séparation des facteurs. L'intégrale se définit donc par :

$$I = I_{F_V} + I_{R_V} = \frac{A}{2Rdh} \left(I_{front}^{cyl} k_v I_{F'_V} + I_{rear}^{cyl} \bar{k}_v I_{R'_V} \right)$$

avec

$$I_{F'_V} = \int_{F_V} 1 - A \int_{F_V} \frac{k_v}{dh} (R_x + z) - A \int_{F_V \times R_L} z_{shad} \frac{\bar{k}_l}{dh} - A \int_{F_V \times F_L} z_{shad} \frac{k_l}{dh}$$

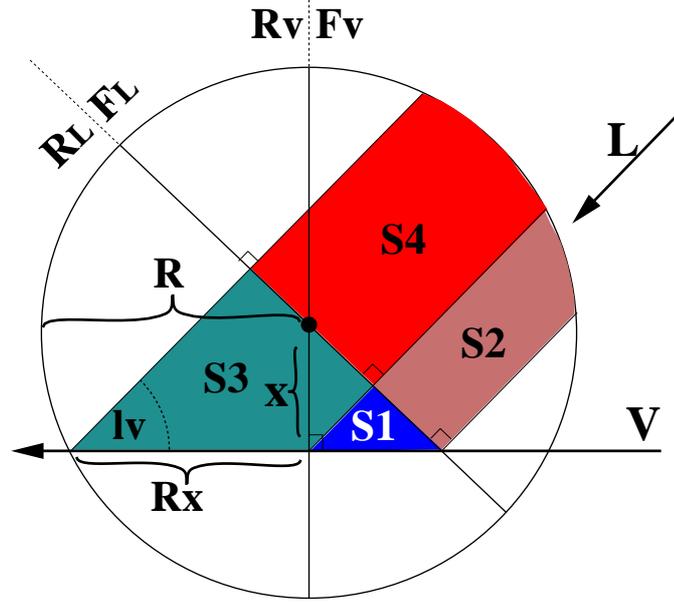


FIG. 4.11 – Le volume de la branche d’aiguille vu par une section orthogonale. Les surfaces des quatre régions $S_1 = F_V \times R_L$, $S_2 = F_V \times F_L$, $S_3 = R_V \times R_L$, $S_4 = R_V \times F_L$ sont proportionnelles à l’intégrale de la longueur du rayon d’ombre, avec l’origine placée sur le rayon (seul le cas générique figure ici). Nous devons intégrer cette surface pour tous les x .

$$I_{R'_V} = \int_{R_V} 1 - A \int_{R_V} \left(\frac{k_v}{dh} R_x + \frac{\bar{k}_v}{dh} z \right) - A \int_{R_V \times R_L} z_{\text{shad}} \frac{\bar{k}_l}{dh} - A \int_{R_V \times F_L} z_{\text{shad}} \frac{k_l}{dh}$$

avec $F_V \times R_L$ la région de R_L couverte par les rayons d’ombres dont l’origine est dans F_V et ainsi de suite pour les autres régions (cf. figure 4.10).

4.5 Intégration géométrique

Nous arrangeons ceci ainsi :

$$I_{F'_V} = \pi \frac{R^3}{2} - A \frac{k_v}{dh} \frac{2}{3} R^3 - A \frac{\bar{k}_l}{dh} \int_{F_V \times R_L} z_{\text{shad}} - A \frac{k_l}{dh} \int_{F_V \times F_L} z_{\text{shad}}$$

$$I_{R'_V} = \pi \frac{R^2}{2} - A \frac{k_v}{dh} \frac{4}{3} R^3 - A \frac{\bar{k}_v}{dh} \frac{2}{3} R^3 - A \frac{\bar{k}_l}{dh} \int_{R_V \times R_L} z_{\text{shad}} - A \frac{k_l}{dh} \int_{R_V \times F_L} z_{\text{shad}}$$

Les quatre intégrales restant dans les formules ci-dessus somment la longueur des rayons d’ombrage qui commencent sur chaque point du rayon et sont inscrits dans la région indiquée en indice, pour chaque rayon incident. Pour le moment, considérons seulement l’intégrale le long du rayon. Le rayon d’ombre balaie une région puisque son origine suit le rayon. L’intégrale de sa longueur le long du rayon principal ressemble fortement à la surface de cette région : elle est proportionnelle à $\frac{1}{\sin(l_v)}$ où l_v est l’angle entre la projection L_P et V_P de L et V dans le plan orthogonal. La “preuve” de ceci est que si P_L est orthogonal à V_P alors l’intégrale de la longueur est l’aire d’une surface. Autrement, on peut se replacer dans ce cas avec un changement de variables ayant pour jacobien $\frac{1}{\sin(l_v)}$. Afin de calculer l’intégrale le long du rayon, nous devons mesurer l’aire de chaque région balayée S_1, S_2, S_3, S_4 en utilisant des relations géométriques et trigonométriques. Nous devons ensuite intégrer le résultat pour chaque rayon. Après de longs et fastidieux

calculs nous obtenons un résultat simple et symétrique (sans approximation) :

$$\begin{aligned}
\int S_1 &= (1 + \cos(l_v)) \frac{R^3}{3} \sin(l_v) \\
\int S_2 &= (1 - \cos(l_v)) \frac{R^3}{3} \sin(l_v) \\
\int S_3 &= (1 + \cos(l_v)/3) R^3 \sin(l_v) \\
\int S_4 &= (1 - \cos(l_v)/3) R^3 \sin(l_v)
\end{aligned} \tag{4.8}$$

Le facteur $\sin(l_v)$ disparaît quand on multiplie par le jacobien.

4.6 Résultat de l'illumination d'un rameau d'aiguilles

L'opacité se déduit facilement :

$$1 - \alpha_{F_V} = \frac{1}{2R} \int_{x=-R}^R A^{R \times \frac{k_v}{dh}} \approx 1 - \frac{AR}{dh} \frac{\pi}{4} k_v$$

i.e.

$$\boxed{\alpha_{F_V} = \alpha k_v, \quad \alpha_{R_V} = \alpha \bar{k}_v} \tag{4.9}$$

Nous introduisons de manière similaire l'opacité dans la direction de la lumière : $\alpha_{F_L} = \alpha k_l$, $\alpha_{R_L} = \alpha \bar{k}_l$ et nous avons finalement $I = I_{F_V} + I_{R_V}$ avec

$$I_{F_V} = I_{\text{front}}^{\text{cyl}} \alpha_{F_V} \left(1 - \frac{8}{3\pi^2} (2\alpha_{F_V} + (1 - \cos(l_v))\alpha_{R_L} + (3 - \cos(l_v))\alpha_{F_L}) \right)$$

$$\boxed{I_{R_V} = I_{\text{rear}}^{\text{cyl}} \alpha_{R_V} \left(1 - \frac{8}{3\pi^2} (4\alpha_{F_V} + 2\alpha_{R_V} + (1 + \cos(l_v))\alpha_{R_L} + (3 + \cos(l_v))\alpha_{F_L}) \right)} \tag{4.10}$$

5 Résultats

Une propriété majeure de ce modèle est l'évolution de son coût en fonction du nombre d'aiguilles, *i.e.* la complexité en fonction du nombre d'aiguilles N par cône et du nombre $\frac{l}{dh}$ de cônes sur une branche par unité de longueur (ces deux nombres sont proportionnels à la racine carrée de la densité d'aiguilles). Si N est multiplié par deux, le nombre d'intersections pour le niveau un et le nombre d'échantillons par pixel qu'un lancer de rayons doit traiter est multiplié par deux, alors que les niveaux deux et trois ne sont pas affectés. Ce raisonnement s'applique si dh est divisé par deux. Le coût des rayons d'ombrage évolue de la même manière. Alors qu'un lancer de rayons classique lance un rayon d'ombre pour chaque échantillon, notre modèle factorise le rayon pour la partie extérieure à la branche.

Nous avons comparé l'efficacité de notre modèle avec un lancer de rayons classique, *Rayshade*, utilisant le sur-échantillonnage pour diminuer l'aliassage. Il est important de savoir que le nombre maximum de rayons lancé par pixel de *Raushade* est de 64. Donc, quand un arbre est loin (c'est-à-dire moins de 100 pixels de haut), *Rayshade* ne lance pas assez de rayons pour éviter l'aliassage.

Il semble efficace en temps de calcul mais c'est au dépend de la qualité. Sur une image fixe comportant beaucoup de hautes fréquences, comme peut l'être une image de forêt, l'aliassage n'est pas toujours visible parce qu'il est difficile de distinguer le bruit de l'information. Par contre, dès que l'on calcule une animation, l'aliassage apparaît immédiatement (grouillement).

Notre scène de test comporte 80 sapins, qui représentent à peu près 127 pixels de haut pour les plus proches et 64 pour les plus lointains (cf. figure 4.12).

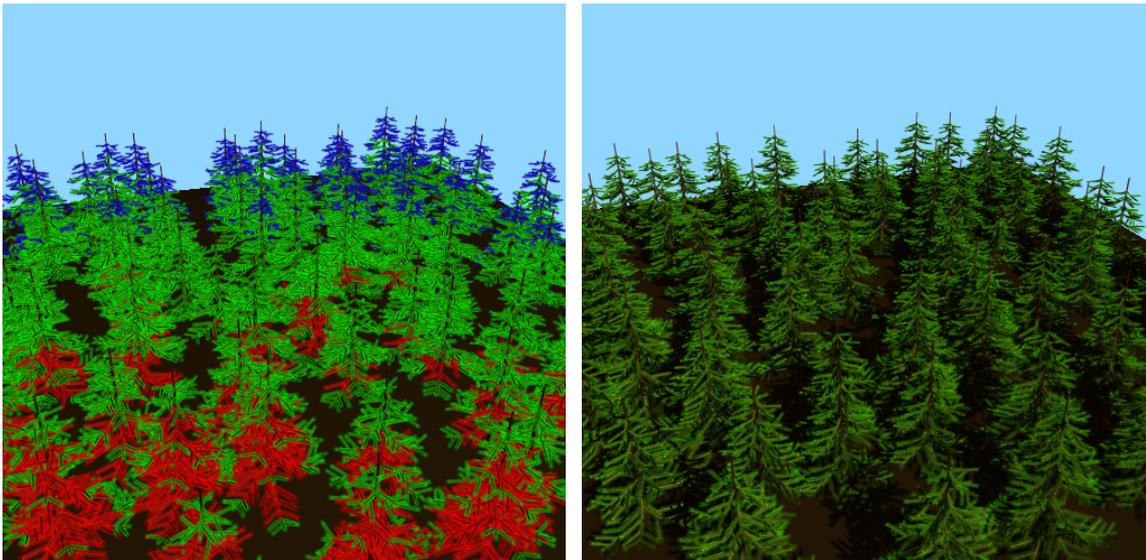


FIG. 4.12 – Notre scène de test. À gauche : Les trois niveaux de détails (niveau un en rouge, niveau deux en vert et niveau trois en bleu). À droite : 80 sapins.

Les sapins utilisés pour nos tests contiennent généralement 300 branches et à peu près 30000 aiguilles, la scène contient donc environ 2 millions d'aiguilles. Concernant une branche, un cône a 3.94 cm de haut et à un rayon de 1.6 cm, une ouverture de $\pi/8$ et le pas entre les cônes est de 0.9 cm. Il y a 12 aiguilles par cône pour ces arbres, qui ont un rayon de 0.05 cm et une longueur de 4.25 cm. En moyenne, 4.4 cônes sont imbriqués, par conséquent un rayon passant à travers l'axe orthogonalement à la branche traversera en moyenne 8.8 couches.

Considérons maintenant les temps de rendu. Les tests de comparaison entre notre modèle et le modèle classique⁴ qui utilise *Rayshade* ont été effectués sur une Onyx² Infinite Reality en exploitant un seul processeur. *Rayshade* a effectué le rendu de l'image 4.12 en 65.3 minutes avec une optimisation à base de grilles. Notre modèle⁵ calcule l'image en 8.1 minutes. Avec une implémentation largement améliorable notre méthode est 8.1 fois plus rapide que le modèle classique représenté par *Rayshade*. Pour un grand paysage où les arbres lointains sont très petits, *Rayshade* ne peut empêcher un fort aliassage du fait de son seuil maximum de 64 rayons par pixel. Si nous pouvions augmenter cette limite, le gain en notre faveur serait largement plus important.

⁴Le système *Rayshade* utilise le sur-échantillonnage pour diminuer l'aliassage, et toutes les aiguilles sont représentées par des cylindres.

⁵Notre implémentation utilise le lancer de cônes (un unique rayon par pixel) pour diminuer l'aliassage et nous utilisons les 3 niveaux de détails.

5.1 Parallélisation de l'algorithme

Malgré l'accélération due à notre modèle, une scène complexe calculée par lancer de rayons devient vite très coûteuse en temps de calcul. Diminuer ces temps de calcul est intéressant dans le cadre d'applications de mise au point ou de rendu interactif : pour un artiste il est commode de disposer d'un aperçu du résultat rapidement. Pour ceci, une des possibilités à étudier est le calcul parallèle.

Nous avons parallélisé [MC01] notre lancer de cônes (*cf.* annexe B) suivant un schéma de parallélisation classique, *i.e.* en découpant une animation ou une image en sous-parties de manière à distribuer les tâches. Notre première machine de test a été une *Onyx*² Infinite Reality avec 6 processeurs (*MIPS R12000* à 400MHz), puis nous avons effectué le portage de l'application sur une grappe expérimentale de PC⁶. Les résultats de cette parallélisation sont quasi-linéaires sur les deux plates-formes : avec 6 processeurs une animation est calculée 6 fois plus vite et une image 5.9 fois plus vite. En performance absolue la grappe de 12 PC est environ 30% plus rapide que l'*Onyx* avec 6 processeurs. La différence se situe donc au niveau du coût financier des deux machines : l'*Onyx* coûte à peu près quatre fois le prix d'une telle grappe. On en conclue que pour une application comme la notre, où l'algorithme est fortement parallélisable, l'investissement dans une grappe de PC est un choix à envisager. Consulter l'annexe B pour plus de détails.

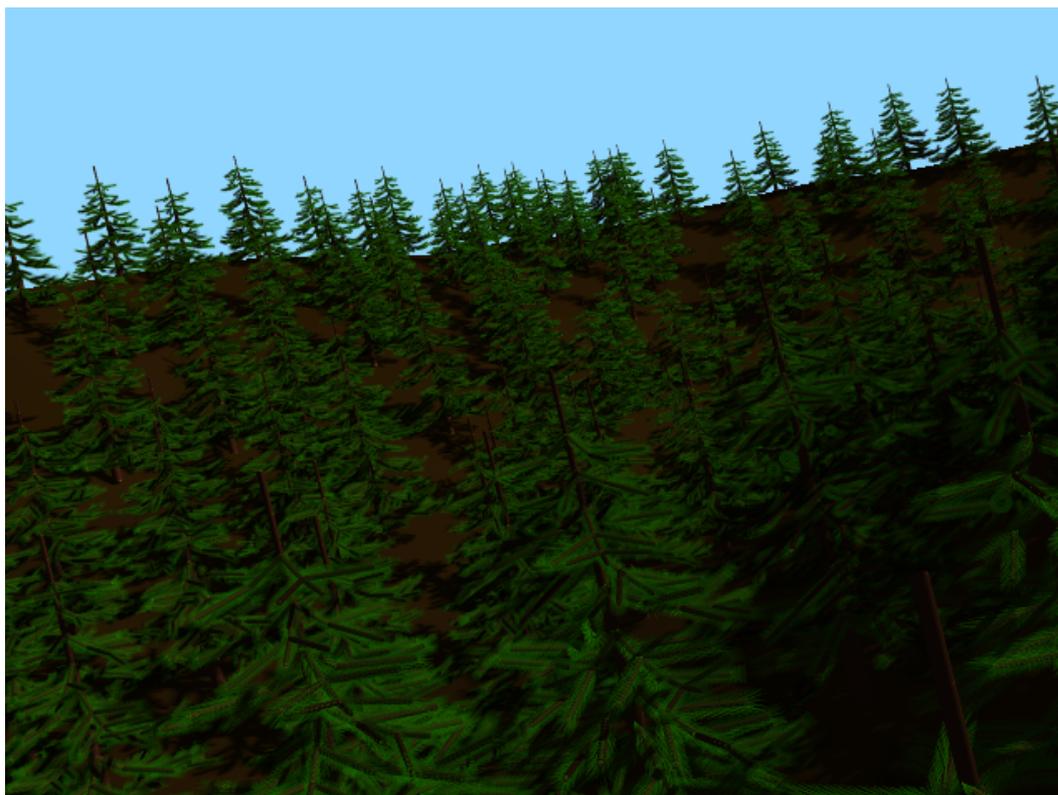


FIG. 4.13 – Une scène de 1000 arbres.

⁶Nous avons utilisé la grappe de l'équipe *SIRAC* composée de 12 PC Pentium II (450MHz) sous Linux, disposant du module expérimentale *SciFS* (developpé par Emmanuel Cecchet) offrant une mémoire partagée distribuée facilitant la gestion de la mémoire pour des applications distribuées.

6 Conclusion et perspectives

Nous avons introduit un ensemble de trois *shaders* capables de représenter à différents niveaux (aiguilles, cônes et rameaux) les effets cumulés des niveaux inférieurs sans avoir à les échantillonner, en tenant compte de l'auto-ombrage interne ainsi que de la visibilité. Comme toutes nos intégrations sont analytiques, nous pouvons produire des images de qualité (en particulier avec peu d'aliasage) et ce, rapidement. D'un point de vue théorique, nous aimerions améliorer les approximations faites. Il serait, par exemple, intéressant de lever l'hypothèse d'un albédo faible en substituant une loi polynomiale à l'approximation linéaire.

Les paramètres des *shaders* nous permettent de simuler différents types de conifères (pins, sapins, etc.) et de moduler les caractéristiques d'un arbre (par exemple pour simuler l'effet du vent). Nous avons été capables d'intégrer ces *shaders* parce les objets de notre étude sont très structurés. Réciproquement, l'usage massif de cette connaissance a priori, fait que ces trois *shaders* ne peuvent simuler que des branches composées d'aiguilles. Dans la nature, de nombreux objets sont composés d'une unique sorte de structure, ou présentent des similarités, il est donc possible d'intégrer analytiquement leur *shader*. La prochaine étape pour nous serait de simuler d'autres types d'arbres, pour lesquels la structure est plus stochastique (concernant la distribution et l'orientation des feuilles). Il serait aussi intéressant de gérer des structures plus grandes en nombres de primitives, comme un ensemble de rameaux, voire un arbre entier.

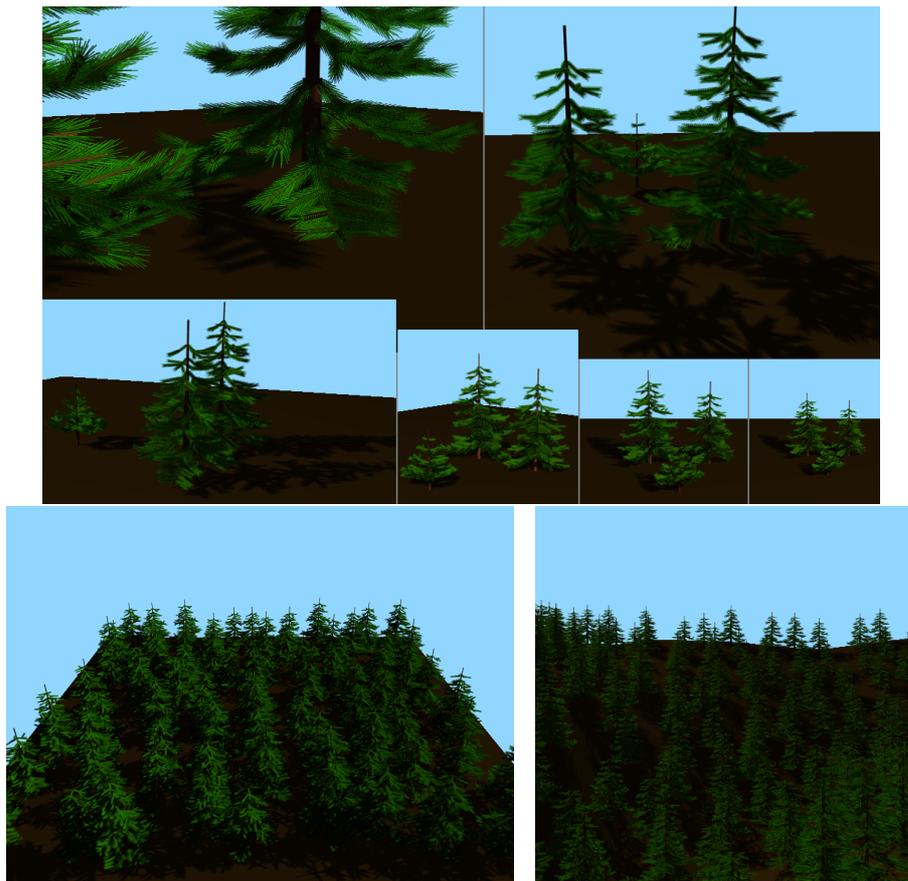


FIG. 4.14 – *En haut* : trois arbres, depuis un point de vue proche jusqu'à un point de vue lointain. *À gauche* : 100 arbres sur un carré. *À droite* : une scène de 1000 arbres.

