

物理シミュレーションに基づく風に揺れる樹木のアニメーション

赤木 康宏[†] 佐波 晶[†] 北嶋 克寛[†]

本研究は、景観シミュレーションなどで用いられる樹木のCGモデルを用い、樹形や葉の大きさの違いによる気流の変化を考慮した風に揺れる樹木のアニメーションを生成するための手法を提案している。複雑な形状を持つ樹木周辺の風のシミュレーションを行うためには、枝や葉などの風を遮る物体が風に及ぼす影響を考慮する必要がある。風の物理シミュレーションモデルにおいて非圧縮性流体におけるNavier-Stokesの方程式を用いると、一般に次のような問題が生じる。樹木形状の詳細な形状までを考慮すると、計算量が増大しリアルタイムのアニメーション生成が困難になる、という問題である。そこで、本論文では、枝や葉のように類似した形状を持ちかつ樹木全体に数多く存在するパーツを風速を減少させる単純な抵抗体としてモデル化し、抵抗体の空間分布を表す境界条件マップを樹木モデルから自動生成することにより、計算量を減らしリアルタイム性のあるアニメーション生成を可能にする新たな手法を提案している。また、本手法は、空間全体の風の動きに階層的な計算手法を適用することにより、樹木周囲の自然な風の流れや樹木どうしの影響などについても高速に計算することができるという特長を持つ。これらの特長に関する各種実験結果を示し、複雑なパーツ形状からなる樹木が風に揺れる様子のアニメーションを力学的な根拠に基づきかつ高速に自動生成できたとを実証した。

Computer Animation of Swaying Trees Based on Physical Simulation

YASUHIRO AKAGI,[†] SYO SANAMI[†] and KATSUHIRO KITAJIMA[†]

This paper presents a series of techniques for generating animations of trees swaying in the wind, in consideration of the influences that the tree shapes and leaf sizes give to the air current. To do the simulation of the wind around a tree having a complicated shape, it is necessary to consider the influence that some objects obstructing the wind such as leaves or branches give. Generally, the following problem occurs when we use the incompressible Navier-Stokes equations in a physical simulation model of the wind. Computational complexity increases because of considering the details of tree shapes, so it is difficult to generate the animations in real-time. Therefore, this paper proposes a novel method that reduces the computational complexity and realizes an animation in real-time, by means of a boundary condition map expressing space distribution of resistances from tree models automatically. In this case, we make a model as simple resistances decreasing the wind velocity from the parts that have similar shapes like leaves and branches. And also, it has another advantage that the influences between a tree and others can be rapidly calculated by using a hierarchical calculation method. Finally, through many experiments using these methods, it is shown that real-time animations of swaying trees in the wind can be realized.

1. はじめに

近年、樹木のCG表現に関する研究の発展により、自然な樹形の自動生成や、生長シミュレーションに関する研究が数多く行われ、本物に近い樹木形状の生成が可能になっている。

静的な樹木形状を表現する研究としては、Weberら¹⁾の研究に代表されるように、枝の角度や長さ、分枝数、幹や枝の接続関係などを用いて樹木形状のデー

タ構造を表現する研究が行われてきた。近年では、このような樹木データに対して、周囲の環境を考慮した生長シミュレーションを適用することで、枝の数や大きさを変化させ、より自然な樹木形状を生成する研究²⁾⁻⁴⁾が行われている。また、写真やレンジスキャナなどを用いた樹木の立体復元に関する研究^{5),6)}などのように、自然界に存在する樹木の形状を取り込むことにも成功している。このように、静的な樹木形状の生成に関する研究においては個体の持つ特徴や周囲の環境を考慮した手法が数多く提案されており、様々な個性を持つ樹木を扱うことが可能になっている。

一方、風による樹木の揺れをはじめとする動的な現

[†] 東京農工大学

Tokyo Agriculture and Technology University

象を扱う研究⁷⁾⁻¹³⁾も、樹木を含むアニメーションに欠かせない要素として重要性が高く、数多くの研究が行われている。Otaら⁷⁾は、枝や葉の変形を風力に応じたモード関数から求める方法をとっている。この手法は、非常に高速に樹木の揺れを求めることができるが、樹木を揺らす原因となる風を物理現象として扱うことは行っておらず、枝や葉の揺れによる気流の変化などを求めることは行っていない。Giacomoら⁹⁾の研究では、風の簡易モデルと樹木の力学シミュレーションを組み合わせ、風と樹木の相互作用を表現した。この方法の特徴は、計算量が少なくすむのでリアルタイム性のあるアニメーションを実現することができるが、樹木の風上側と風下側との気流の変化が考慮されていないことや、他の樹木からの影響による局所的な風の強弱が表現できない、などの問題がある。

これらの方法に共通する問題は、風の物理シミュレーションを行っていないことに起因している。しかし、物理シミュレーションを行えば、樹木の存在する空間を流れる風を詳細に計算することができるが、過去の流体力学を用いた様々なアニメーションに関する研究¹⁴⁾⁻¹⁷⁾から類推されるように、膨大な計算時間を必要とするために、短時間で解を求めることが難しくリアルタイム性が失われてしまうという問題がある。本研究は、風の物理シミュレーションを行うことを特徴とし、計算の高速化を図るための様々な工夫を行うことにより、樹木の持つ枝や葉の形状や特性を反映した揺れをリアルタイムのアニメーションとして生成する一連の手法を構築することを目的とする。

2. 風の物理シミュレーションモデル

本章では、境界条件マップなどの流体力学に関連する新たな手法を提案するための前提として必要になる、格子法を用いた数値流体シミュレーションについて述べる。流体力学では、風速による圧縮や温度などの条件によって計算に用いる運動方程式が異なる。本研究では自然界で発生する様々な風のうち、木を揺らす程度の風を対象とする。このような風は音速に比べてきわめて遅く、圧縮が起きず、アニメーション時間において温度の急激な変化が起きないと仮定しても差し支えない。このような条件を考慮し、非圧縮性流体における Navier-Stokes の方程式を用いる。

2.1 Navier-Stokes の運動方程式¹⁸⁾

3次元流れのシミュレーションを行う際に用いる、非圧縮性流体における Navier-Stokes の方程式は次のように表される。

$$\frac{\partial \mathbf{V}}{\partial t} + \nabla \cdot (\mathbf{V} \cdot \mathbf{V}) = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{V} \quad (1)$$

t : 時刻, $\mathbf{V} = (u, v, w)$: 流速ベクトル

p : 圧力, ρ : 密度, ν : 動粘性係数

$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$: ナブラ演算子

式(1)は、流速 \mathbf{V} の時間変化を表す式である。また、非圧縮性流体では流体の密度が変化しないので、特定の空間に流入する物質の量と流出する量が等しくなければならない。そこで、流れ場全体の \mathbf{V} についての質量保存の法則を表す連続の式(式(2))を満たす必要がある。

$$\nabla \cdot \mathbf{V} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (2)$$

式(1)、(2)を解くことにより、3次元空間での風速の変化が求められる。

2.2 Navier-Stokes の方程式の数値解法

式(1)、(2)は非線形方程式であり、厳密解はきわめて簡単な条件の流れに対してしか得られない。そこで、空間を有限の格子に分割し離散化を行うことで数値解を求めることが一般的に行われている。本研究では、スタガード格子¹⁹⁾を用いた空間分割により流れ場を離散化し、SMAC法²⁰⁾を用いて差分方程式を解くことにより各格子の風速を求める方法を用いる。

2.2.1 スタガード格子による流れ場の表現

本項では、本研究で用いたスタガード格子による流れ場の離散化法について述べる。式(1)、(2)で示した Navier-Stokes の方程式には各方向の流速、圧力、密度および動粘性係数が含まれるが、このうち密度、動粘性係数は定数として与えられるので、空間全体の流速および圧力を未知数とする方程式を解くことにより、風の時間変化が求められる。この流速と圧力を格子のどこで考えるかに、スタガード格子の特徴がある。単純な格子法では、各格子における圧力と流速を格子の中心部分で代表させるのに対して、スタガード格子では、流体の出入りを格子の6つの境界面で代表させる(図1)。流速を扱う位置を格子の中心から境界面上に移すことにより、差分をとる際に、隣接する圧力と流速の位置が近づき精度が高まる、という特長がある。また、6つの面の流速の違いから流れに圧力差が生じたり、また圧力差から各方向の流速が変化したりするなどの現象を扱うことができる。

2.2.2 SMAC 法

SMAC法は、スタガード格子を用いて離散化したモデルに対する数値解法の1つであり、数値流体力学で広く用いられている。本項では、SMAC法の概要

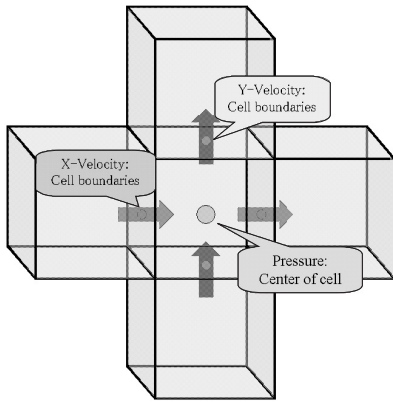


図 1 スタガード格子
Fig. 1 Staggered cell.

について述べる。

式 (1) に関して、ある時刻 t における既知の流速 \mathbf{V} を \mathbf{V}^t とし、次の時刻 $t + 1$ における未知の流速 \mathbf{V}^{t+1} とすると、次式のように表される。

$$\mathbf{V}^{t+1} = \mathbf{V}^t + \Delta t \left(-\frac{1}{\rho} \nabla p^{t+1} - \nabla \cdot (\mathbf{V}^t \mathbf{V}^t) + \nu \nabla^2 \mathbf{V}^t \right) \quad (3)$$

式 (3) の右辺のうち、 $\nabla \cdot (\mathbf{V}^t \mathbf{V}^t)$ (移流項) と $\nu \nabla^2 \mathbf{V}^t$ (拡散項) は既知であるが、圧力 p^{t+1} は未知である。式 (3) に対して両辺の発散 (divergence: ∇) をとると次式が得られる。

$$\frac{\nabla \cdot \mathbf{V}^{t+1} - \nabla \cdot \mathbf{V}^t}{\Delta t} = -\nabla^2 \left(\frac{p^{t+1}}{\rho} \right) + \nabla \cdot (-\nabla \cdot (\mathbf{V}^t \mathbf{V}^t) + \nu \nabla^2 \mathbf{V}^t) \quad (4)$$

\mathbf{V} はすべての時刻において式 (2) を満たす必要があるが、数値計算においては近似や丸めによる誤差が生じるので、現在の時刻において求まっている $\nabla \cdot \mathbf{V}^t$ の値は一般に 0 とはならない。時間進行による誤差の累積を抑えるために、 p^{t+1} を求める際に $\nabla \cdot \mathbf{V}^t$ の値をあえて残し、流れ場の理想状態である $\nabla \cdot \mathbf{V}^{t+1} = 0$ を満たすような解を求める。式 (4) において $\nabla \cdot \mathbf{V}^{t+1} = 0$ として変形すると、次式が得られる。

$$\nabla^2 \left(\frac{p^{t+1}}{\rho} \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{V}^t + \nabla \cdot (-\nabla \cdot (\mathbf{V}^t \mathbf{V}^t) + \nu \nabla^2 \mathbf{V}^t) \quad (5)$$

式 (5) の右辺は時刻 t における既知の項のみであるので式を解くことができるが、格子数を n とすると

n 元の連立差分方程式となる。一般に数百元を超える連立方程式は反復解法を用いて解くので、打ち切りなどによる誤差が生じる。したがって、式 (5) から得られる圧力 p^{t+1} には誤差が含まれ、計算を繰り返すことで誤差が蓄積される。そこで、既知の圧力 p^t の持つ誤差を考慮した計算を行うために、 $\mathbf{V}^{t'}$ を用いて式 (3) を次のように分割する。

$$\mathbf{V}^{t'} = \mathbf{V}^t + \Delta t \left(-\frac{1}{\rho} \nabla p^t - \nabla \cdot (\mathbf{V}^t \mathbf{V}^t) + \nu \nabla^2 \mathbf{V}^t \right) \quad (6)$$

$$\mathbf{V}^{t+1} = \mathbf{V}^{t'} - \Delta t \nabla \phi \quad (7)$$

$$\phi = \frac{p^{t+1} - p^t}{\rho} \quad (8)$$

式 (6) の右辺は、Navier-Stokes の方程式を陽解法的に差分化した形となっており、時刻 $t + 1$ における流速を概算したものに相当するが、陽解法による近似および p^t に由来する誤差が含まれる。式 (7) は $\mathbf{V}^{t'}$ に対してステップ間の圧力差 ϕ による誤差の補正を加え、Navier-Stokes の方程式を満たす \mathbf{V}^{t+1} を求める計算を行っている。式 (7) において両辺の発散をとると、式 (5) と同様に整理すると次式となる。

$$\nabla^2 \phi = \frac{1}{\Delta t} \nabla \cdot \mathbf{V}^{t'} \quad (9)$$

式 (9) の $\mathbf{V}^{t'}$ は既知であるので、これを解くことで ϕ の値が求まるが、この解法には、次式に示す格子点 (i, j, k) における差分方程式を用いる。

$$\begin{aligned} & \frac{\phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k}}{\Delta x^2} \\ & + \frac{\phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k}}{\Delta y^2} \\ & + \frac{\phi_{i,j,k-1} - 2\phi_{i,j,k} + \phi_{i,j,k+1}}{\Delta z^2} \\ & = \frac{1}{\Delta t} \left(\frac{-u_{i-\frac{1}{2},j,k}^{t'} + u_{i+\frac{1}{2},j,k}^{t'}}{\Delta x} \right. \\ & \quad + \frac{-v_{i,j-\frac{1}{2},k}^{t'} + v_{i,j+\frac{1}{2},k}^{t'}}{\Delta y} \\ & \quad \left. + \frac{-w_{i,j,k-\frac{1}{2}}^{t'} + w_{i,j,k+\frac{1}{2}}^{t'}}{\Delta z} \right) \quad (10) \end{aligned}$$

この ϕ に関する差分方程式は、SOR 法などの反復解法により解くことができる。 ϕ が求められることは、式 (5) によって p^{t+1} を直接求める方法に比べ、誤差を打ち消す効果がある。式 (8) から、 $p^{t+1} = \rho\phi + p^t$ として p^{t+1} が求められる。これを式 (7) に代入して \mathbf{V}^{t+1} が求まるので、 \mathbf{V}^{t+1} の累積誤差も抑えられる。

2.2.3 計算格子の境界条件

空間を有限の格子に分割すると、外周の格子は片側の情報が得られなくなるので、そのままでは差分法を用いた計算を行うことができない。このため、境界部分でも流れが連続になるように特別な処理を行う必要があるが、本研究では、流出境界に Sommerfeld の放射条件を、地面などの固体壁には No-Slip 境界条件を用いる。

2.3 樹木が風に与える影響の扱い

2.2 節で述べた手法は、空間内に樹木が存在しないことを前提としていた。空間内に樹木が含まれる場合には、枝や葉が風の抵抗となって働く現象をモデル化しなければならない。これまで用いてきた Navier-Stokes の式は外力を含まない式であったが、本研究では、この式に外力項を加えることによって、この現象を扱う。すなわち、ある格子に枝や葉が含まれている場合、風速 V^i を求めるための式 (6) の右辺に、それらの影響に応じた外力項を加える。このとき、樹木の各パーツが風に及ぼす影響を個々のパーツレベルで考えるのではなく格子単位で考え、この風と樹木の抵抗との相互作用を表現する。この対応関係を定義した情報を境界条件マップと呼ぶ。このような境界条件マップを用いることで、格子サイズを大きくしても、枝や葉の集合を風に対する 1 つの抵抗体として扱うことが可能になり、計算の高速化を図ることが可能になる。これについての詳細は、3.2 節で述べる。

3. アニメーションの生成手法

本章では、対象とする樹木の力学モデルおよび風の物理シミュレーションに基づく樹木の揺れを表現する一連のアニメーション生成手法について述べる。はじめに、3.1 節で樹木モデルの構造および風力による枝の曲げを求める手法について述べる。次に、3.2 節で境界条件マップによる風のシミュレーションの高速化について述べる。最後に、3.3 節でさらに計算時間を短縮するために併用するその他の高速化手法について述べる。

3.1 アニメーションに用いる樹木モデル

景観シミュレーションや風に揺れる樹木のアニメーションなどで用いられる樹木モデルの多くは、枝や葉などの各パーツのサイズ、角度、親子関係によって形状が表現されている。また、植物学の分野においても、樹木を枝や葉などのパーツ単位で扱い、樹木形状の生成モデルの仮説を検証するために、同様の研究が行われている²¹⁾。そこで、本研究ではこれらの研究を参考にして、1 本の幹を基準とし、親子関係を持つ枝や葉

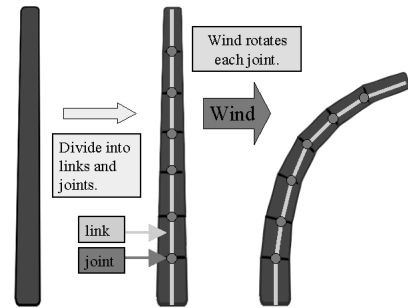


図 2 枝の曲げ

Fig. 2 Bend of branch.

を再帰的に生成することにより樹木形状を構成する方法をとる。なお、前述した樹木形状の生成に関する研究においては生長シミュレーションを行うことを主眼としているが、本研究の目的は異なるので、複雑な生長シミュレーションなどは行わず、単純な生成規則によって生成する。なお、一般的なパーツと構造を用いているので、他の研究で生成された樹木モデルを用いることも可能である。

3.1.1 枝の力学モデル

上述したように、CG で広く用いられる樹木モデルは単純な構造が再帰的に繰り返す構造になっている。このような単純なモデル上での風によって生じる枝の変形は、風力と枝のしなりとの関係とパーツ間の力の伝播を定義することにより、比較的容易に扱うことが可能になる。本研究で用いる樹木モデルでは、ある枝とその子の枝または葉との接続部分を接点（固定部分）と呼び、また 1 本の枝をいくつかの円錐台状のパーツ（リンク）に分割しその接続部分は節点（ジョイント）と呼ぶ。各リンクは剛体とし、節点のみ回転の自由度を持つ。したがって、枝のしなりは各節点の回転によって表現される（図 2）。

なお、力学的な計算のことだけを考えた場合には、枝をはりのモデルで近似し解析的に計算を行ってもそれほど手間はかからない。しかし、本研究は、リアルタイムアニメーションを行うことを目的としているので、表示の際の計算が高速に行えるモデルでなければならないことを同時に考慮する必要がある。このため、このようなモデルを採用している。各節点の回転角は、図 3 の左に示すように、着目する節点より先端側にある節点にかかる力が着目する節点に与えるモーメントの総和を計算することにより、求められる。このとき、モーメントと各節点の回転角との関係については、Terashima らによる樹木の力学バランスに関する研究²²⁾ の中で定義された枝令（枝の年齢）と弾性係数との関係式を用いている。すなわち、枝の先端に

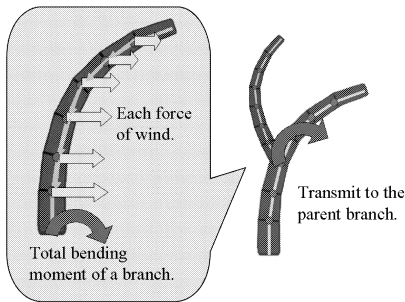


図3 枝にかかる力の伝播

Fig. 3 Transmission of a force added to a branch.



図4 樹木モデル

Fig. 4 Tree model.

行くほど枝令が若く弾性係数が高くなるので、先端の節点ほど回転角が大きくなる。子の枝から親の枝への力の伝播は、親の枝との接点において行われ、その力は親の枝の節点に直接かかる風力に加えられる（図3の右）。なお、本研究では初期の枝形状は重力との釣り合いのとれた状態とし、枝のしなりを求める際には、重力の影響は無視し風力のみを考慮する。

3.1.2 葉の力学モデル

次に、葉の力学モデルについて述べる。葉は枝とは異なり、大きな空気抵抗を受けるので、葉柄（葉と枝の間の軸）が大きく変形し風の運動方向に沿って全体が大きくなびく。そこで、葉に関しては、葉自体の曲げは表現せずに剛体として扱い、親となる枝との接点を節点（ジョイント）として扱い、その節点における回転角を、葉にかかるモーメントと葉柄の弾性係数によって決定する。

3.1.3 樹木の自動生成

著者らは、種の特徴に基づく樹木形状の自動生成の研究²³⁾を別途行っているが、本研究においては、代表的な樹木を想定し本研究の有効性を検証することとする。代表的な樹木の構造として、ケヤキに似た図4のような構造からなる樹木を採用する。この樹木は、

表1 格子数と計算時間

Table 1 Calculation time of the wind with a grid size.

grid size	calculation time (sec.)
10 ³	0.0016
20 ³	0.12
30 ³	1.13
40 ³	7.08
50 ³	28.74
60 ³	65.81

1つの枝に対して8つの子の枝または葉を持つ構造を再帰的に4回繰り返して自動生成される。自動生成する際には、枝の角度などにばらつきを持たせることにより、個体差を表現している。

3.2 樹木モデルの影響を考慮した風の物理シミュレーションとその高速化手法

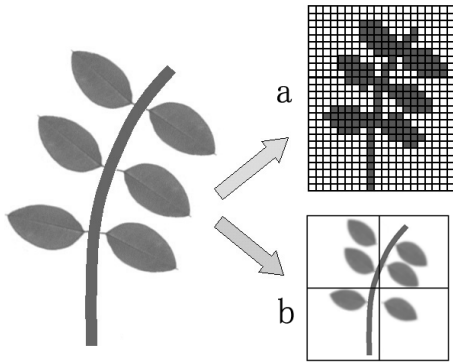
次に、2章で述べた風の物理シミュレーションと3.1節で述べた樹木モデルとの相互作用に基づいた風に揺れる樹木のアニメーションを自動生成する手法について述べる。

3.2.1 空間分割と計算時間

風と樹木との相互作用については、2つの側面がある。1つは、風の物理計算を行う際に、樹木の枝や葉が風にどのような影響を及ぼすかであり、もう1つは、その風の影響を受けた樹木の枝や葉がどのようにしなる（変形する）かである。本研究では、前者の計算をできるだけ正確に行うことを主眼としており、後者の計算については、前者の計算に基づいて枝などにかかる外力の分布を変更するだけにとどめ、風に対する一般的な変形のモデルで対応する。

したがって、前者の計算が重要となるが、この際、枝や葉などが風をどのように遮るかを計算格子にどのように反映させるのが最大の問題となる。一般に、葉1つ1つの形状の違いを風のシミュレーションに反映させようとするれば、空間を葉のサイズよりも十分小さく細分割し、膨大な量の格子を用いなければならなくなり、計算時間の飛躍的な増大を招く。

このことを検証するために、6m立法の空間内に1本の樹木を配置し、格子サイズを変えてSMAC法による計算時間の予備比較実験を行った。この実験で最も分割数の多い60³の分割（10cm立法の格子）とした場合には、1ステップあたり1分以上の計算時間が必要になり（表1）、この時点ですでにリアルタイムのアニメーション生成からはほど遠い計算時間となっている。したがって、これ以上の細分化した格子サイズでの実験は打ち切った。この実験から分かるように、リアルタイム性のあるアニメーションを生成するためには、より粗い格子を用いて、かつ枝や葉の気流へ及



a: Ordinary grids and boundary conditions to compute the wind.
b: Using virtual resistance to reduce the amount of grids.

図5 仮想的な抵抗
Fig. 5 Virtual resistance.

ばす影響を扱うことのできる手法が必要になる。そこで、本研究では、粗い格子内でもその中に存在する枝や葉の物理的特性が風の計算にうまく反映できるように、境界条件マップなるものを定義し、計算時間の短縮を図る。

3.2.2 境界条件マップ

計算時間の短縮を行うために、枝や葉などの、樹木全体に数多く存在する似た形状を持つパーツに着目する。特に葉は、樹木全体にほぼ同じような形状が数多く分布しているので、これらを同じ特性を持つ仮想的な抵抗体と見なせば、計算格子を粗くしたとしても、樹木の影響を十分に反映させることができる(図5)。そこで、図5の右下に示すように、粗い格子を設定し、各格子内に存在する葉の集合を仮想的な1つの抵抗体として扱う。こうすることで、葉の気流への影響をラフに考慮した計算が行え、計算量を大幅に減少させることができる。枝についても、同様な考えに基づき、太さが格子サイズ以下の場合には仮想的な抵抗として扱う。ただし、太さが格子サイズ以上の場合には、その枝を含む格子を通常の障害物として計算を行う。

以上のように、樹木のパーツを計算格子内に含まれる仮想的な抵抗と見なし、格子サイズによらずパーツと計算格子の対応関係を表現する方法を境界条件マップと呼ぶ。この境界条件マップを用いることで、樹木形状の変形にともなう流れ場の変化を、マッピング情報の変更という手続きのみで対応することが可能となり、風力の樹木へのフィードバックにも用いることができる。

3.2.3 葉の抵抗モデル

本項では、境界条件マップに用いる葉の抵抗モデル

について述べる。葉は弾性力が低いので、風を受けた場合には、葉全体が流れと同じ方向に引っ張られる傾向がある。葉は、枝に比べ重量が軽く表面積が大きい。そこで、風の影響を強く受け、瞬間的には風速と同じ速度で移動するものとする。一方、葉は枝からの力によってその場にとどめられ、これが風への抵抗力となる。そこで、質量 M の葉が風に及ぼす抵抗力 F_l を、次式のように設定する。

$$F_l = -M \frac{V^t}{\Delta t} \quad (11)$$

F_l を式(6)に外力として与え、SMAC法の計算を行うことにより、葉の影響を物理シミュレーションへ反映させることができる。なお、葉は樹木の中で様々な方向を向いているので、厳密には、風に対する抵抗力はそれぞれの葉によって異なる。しかし、本研究ではCGアニメーションとして表現することが目的であり、そのような微細な変化は表示上の変化として現れにくいと考え、アニメーション生成の高速化のために、簡便な方法で抵抗力をモデル化している。

3.2.4 枝の抵抗モデル

枝の抵抗モデルとしては、まず枝の太さが1つの格子サイズよりも太い場合にはその格子を固体壁として扱い、No-Slip境界条件を用いる。枝が格子サイズよりも細い場合にはその体積に応じて流量を制限する抵抗体として扱う。格子を通過する風は枝の隙間をぬって流れるので、両者の体積の比率に応じて風が制限されると考える。格子の体積を S_c 、枝の体積 S_b とするとき、その格子を通過する流速は、次式のように減速するものとする。

$$\frac{\partial V^{t+1}}{\partial t} = \frac{\partial V^t S_b}{\partial t S_c} \quad (12)$$

枝を含む格子では、風の計算の際に式(12)を用いて流速 V^{t+1} を求める。

3.2.5 境界条件マップを用いた処理の流れ

本項では、境界条件マップを用いた処理の流れを示す。

- (1) 樹木形状から境界条件マップを生成する。
- (2) SMAC法による風の物理シミュレーションを行う。
- (3) 境界条件マップを用いて、式(6)を計算する際に外力項として抵抗力を与える。
- (4) 更新された風速を樹木の各パーツにかかる力として樹木モデルにフィードバックする。
- (5) 枝および葉にかかる風力を用いて、枝および葉のしなりを計算する。

以上の処理を繰り返すことで、風に揺れる樹木のA

アニメーションを生成する。

3.3 アニメーション生成に関する他の高速化手法の適用

3.3.1 階層的な計算手法による高速化

樹木を対象とするアニメーションにおいて、一般に、対象となる空間全体に樹木が密生しているとは限らない。広い空間に樹木がまばらに配置されている場合は、多くの格子では対象物が存在せず、このため気流の変化が起きず、格子を細分化する必要がない。階層的な計算手法では、空間全体をまず粗い格子に分割し、樹木を含む格子については、樹木の大きさに応じて細分化する(図6)。このとき、1つの樹木が複数の粗い格子にわたって存在する場合は、複数の粗い格子を1つに統合し、その統合された領域について上記の細分化を行う。階層間の風力の伝播は、細分化した領域の左右上下のそれぞれの外部境界にあるすべての格子に粗い格子の左右上下のそれぞれの外部境界の風速の値を一樣に与え、詳細な格子で計算を行った後に、左右上下のそれぞれの外部境界のすべての格子について求められた風速の値の平均値を粗い格子に戻すことによって行われる。この手法を用いることで、広大な空間を計算する際に、少ない格子数でシミュレーションを行うことができる。

3.3.2 Level of Detail (LOD) による高速化

LODとは、オブジェクトを視点からの距離や画面上での大きさに応じて、簡易な形状に置き換えたり、自動的に形状を簡略化したりするなどして、表示の処理にかかるコストを減らすための手法全般を指す。本研究では、アニメーション生成の高速化のために、このLODの考え方を風の物理シミュレーションに適用する。人の目には、遠くにある樹木は、詳細な形状が判別できなくなるだけでなく、詳細な動きの変化も判

別できなくなる。そこで、対象とする空間全体の中心と視点との距離が離れるにつれ格子サイズを粗くすることにより、計算の高速化を図る。ただし、形状の簡略化に関するLODと同様に、視点の移動によって格子サイズが動的に変化するので、そのたびに格子の初期化が行われ気流の計算結果に影響を及ぼし、アニメーションの連続性が失われる、という問題が生じる。そこで、この問題に対しては、形状の簡略化に関するLODと同様に、視覚的な変化が現れないように、格子サイズの切換えの際に用いられる視点からの距離を表すパラメータの閾値を実験的に決めておく必要がある。閾値は、表示速度と表示精度のトレードオフによって決められ、どちらを優先させるかは、ユーザの判断で決められる。

4. 実験と考察

本章では、3章までに述べてきた様々な手法を実装し、実装されたシステムによって初めて可能になる気流計算に基づく枝のしなりや葉の揺れに関する様々な実験を行い、考察を加える。実験対象となる樹木モデルは約25,000個のパーツ(枝および葉)からなるモデルを用い、すべての実験で共通に用いる。また、実験における風は、すべての実験に共通に左から右に水平に吹くものとする。本実験で使用した実行環境を、表2に示す。

4.1 樹木の変形

本節では、1本の樹木と風の相互作用の様子が自然に表現されているかを検証するために、2つの実験を行う。はじめに、風速を変化させたときに、1本の枝のしなり方が変化する様子と樹木全体が揺れる様子を検証する。次に、葉の大きさの違いによる気流の変化の違いが、枝のしなり方と樹木全体の揺れの様子へ与える影響を検証する。

4.1.1 風力による樹木のしなり方の変化

樹木モデルに対して3種類の異なる風速を与え、しなり方の変化を比較する。まず、1本の枝の変形を図7に示す。なお、見やすくするために葉を削除して表示しているが、実験は葉がついた状態でやっている。これを見ると、風力に応じてしなりが大きくなる様子が表現されていることが分かる。次に、樹木全体に対し

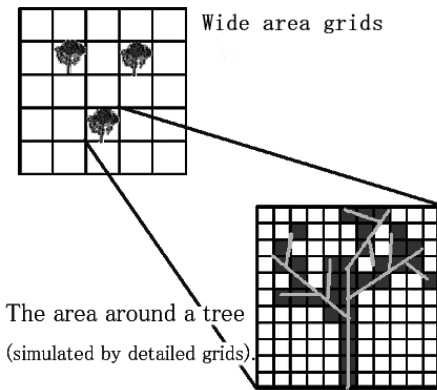
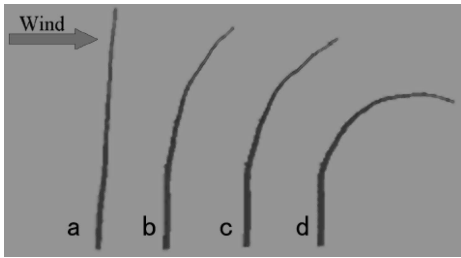


図6 階層的な計算
Fig. 6 Hierarchical calculation.

表2 実験環境

Table 2 Specification of the experiment environment.

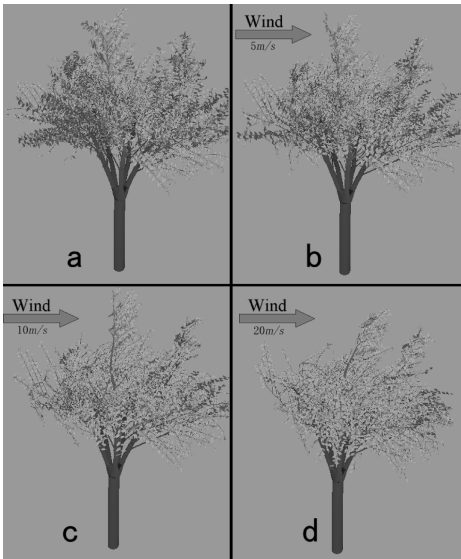
CPU	Pentium4-2.53 GHz
Memory	512 Mbyte
VGA	Radeon9600
OS	Windows2000



Wind velocity (a: 0 m/s, b: 5 m/s, c: 10 m/s, d: 20 m/s)

図 7 風速による枝のしなり方の変化

Fig. 7 Variations of the bend of a branch by each wind velocity.

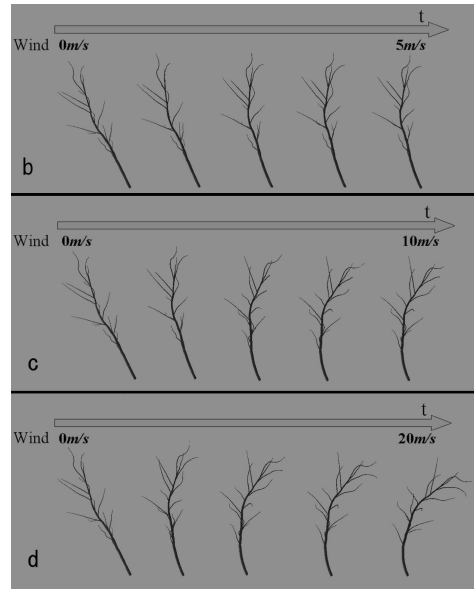


Wind velocity (a: 0 m/s, b: 5 m/s, c: 10 m/s, d: 20 m/s)

図 8 風速による樹木全体の揺れ方の違い

Fig. 8 Differences of the sway of a tree by each wind velocity.

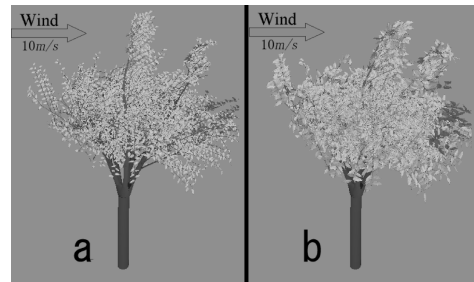
て 3 種類の風速を与えてから十分時間が経過した後の樹木全体の揺れの様子を、図 8 に示す。また、図 8 の各状態において 1 本の直立した枝に着目した際の、枝のしなり方の時間変化を、図 9 に示す。樹木全体に見られる変化は、図 8 の b (5 m/s) では全体がわずかなびく程度だが、図 8 の c (10 m/s) では風を受けている左側の細い枝が大きくしなり、図 8 の d (20 m/s) では太い枝も含めた左側全体が大きくしなっていることが分かる。どの風速においても、風下側に行くほど枝のしなり方が小さくなり、枝や葉が風の抵抗となっている様子が分かる。図 9 に示す枝のしなりの時間変化からは、風速の増加に応じた枝のしなりの増加が表現されていることが分かる。この結果から、風と樹木の相互作用が自然に近い状態で表現されていることが分かる。ただし、枝や葉の間の自己干渉計算が行われ



Wind velocity (b: 5 m/s, c: 10 m/s, d: 20 m/s)

図 9 図 8 の各風速に至る過程での枝のしなり方の時間変化

Fig. 9 Motion of a branch in each tree of Fig. 8.



a: Normal size b: Large size

図 10 葉の大きさによる樹木全体の揺れ方の違い

Fig. 10 Difference of the sway of a tree at each leaf size.

ていないので、大きな変形が生じるケースでは、内側に折りたたまれすぎの現象が見られる箇所もある。

4.1.2 葉の大きさによるしなり方の変化

次に、葉の大きさの違いが樹木全体の揺れと枝のしなりに与える影響について実験を行った。図 10 の a, b の 2 本の樹木は葉のサイズのみが異なり、他の幾何構造と力学的性質については同一とする。a は標準サイズの葉からなり、b は表面積を 2 倍、質量を 4 倍にした葉からなる。図は、同一の風速に対する樹木全体の揺れ方の違いを示す。また、図 10 の a, b それぞれの樹木の 1 本の直立した枝に着目した際の、枝のしなり方の時間変化を図 11 に示す。この結果から、葉の質量が増加した場合 (図 10 の b) に、式 (11) によって葉の受ける風力が増大し、標準サイズの葉からなる樹木 (図 10 の a) に比べ、全体が大きく変形してい

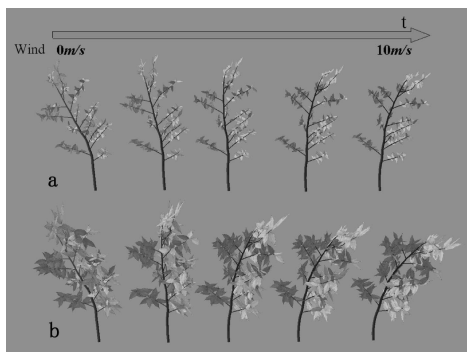
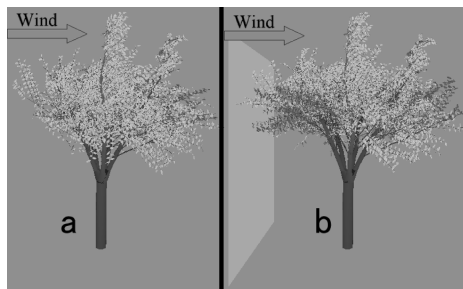


図 11 大きな葉を持つ枝のしなり方の時間変化
Fig. 11 Motion of a branch with large leaves.



a: No wall b: Wall is placed

図 12 風上に固体壁を設置した場合の樹木全体の揺れ方への影響
Fig. 12 Influence of a wall on the sway of a tree, when it is placed on the upwind side.

ることが分かる．枝の動きに関しては，葉のサイズを大きくした場合では曲げが大きくなり，葉の受ける力が增大している様子が分かる．また，葉の大きい樹木では気流に対する抵抗力が増大するので，下流側の風速の平均値が a の樹木の実験に比べ約 1/3 に減少し，気流に対しても葉のサイズが影響を及ぼしていることが確認できる．

4.2 他の物体からの影響

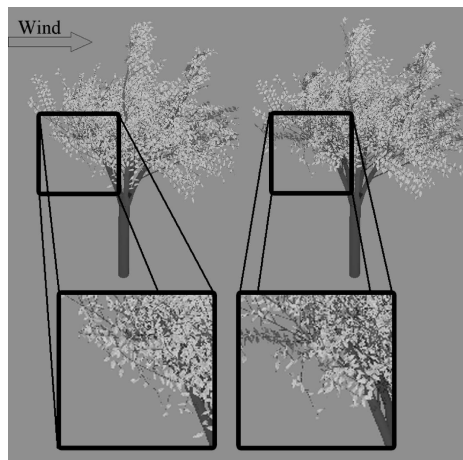
4.1 節では，1 本の樹木についての各種実験結果を示した．本節では，風を遮るような他の物体の存在がアニメーションにどのように反映されているかについて検証する．

4.2.1 固体壁の影響

まず，建物のような変化しない固体壁の影響について実験を行う．図 12 の a は壁のない状態，b は樹木の風上側に風を遮る壁がある状態を表す．両者に同じ風速の風を与えたところ，壁の存在する b の樹木で，壁に近い左下部分において風が遮られ，枝の動きが発生しなくなった．しかし，樹木の中心付近からは，壁を回り込んだ風の影響により枝の揺れが発生している様子が分かる．この結果から，固体壁による風の遮蔽と回り込みが再現されていることが確認できた．

4.2.2 他の樹木の影響

次に，2 本の樹木を同じ空間内に配置し，樹木間の相互作用による揺れの違いを検証する．実行結果を図 13 に示す．同一空間に 2 本の樹木を配置した場合は，風下側（図 13 の b）の樹木の受ける風が，風上側に比べて弱まり，図 13 の拡大部分のように，風上側の樹木（図 13 の a）の陰になる部分でしなり方が小さくなっていることが分かる．この結果から，風の物理シミュレーションを行わない従来の方法では実現が難しいとされている，他の樹木の影響による風の回り込みや風速の増減などが自然に表現できているこ



a: Upper wind b: Lower wind

図 13 風上の樹木から受ける枝のしなり方への影響
Fig. 13 Influence from the tree on the upwind side on the difference of the sway of branches.

とが確認できた．

4.3 格子サイズと計算量

本節では，シミュレーション対象となる空間の分割数，格子サイズを変更したときの，計算量および格子の粗さの違いによる気流変化に起因する樹木の揺れおよびその一部の枝のしなり方の違いについて検証する．

4.3.1 境界条件マップに基づく物理シミュレーションを用いたアニメーション生成に要する計算時間

はじめに，その他の高速化手法を用いずに，境界条件マップのみを用いたアニメーション生成に要する計算時間を計測する．この計算には，SMAC 法の計算に加え，境界条件マップの生成および枝や葉の揺れの計算に要する時間も含まれる．計測結果を表 3 に示す．この結果，格子数が 15^3 ならば，12fps 前後のフレームレートが確保できることが分かる．また，この結果を，表 1 で示した SMAC 法の計算のみに必要な

表 3 格子サイズの違いによるアニメーション生成時間の比較
Table 3 Creation time of the animation at each grid size.

grid size	BCM (sec.)	SMAC (sec.)
10^3	0.062	0.0016
15^3	0.078	0.014
20^3	0.156	0.12
25^3	0.469	0.41
30^3	1.203	1.13

BCM: Boundary Condition Map

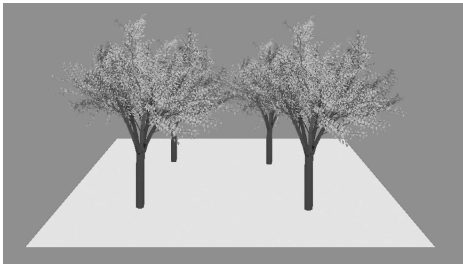


図 14 高速化手法の検証

Fig. 14 Experimental verification of the acceleration technique.

表 4 各ケースにおける計算時間

Table 4 Calculation time in each case.

Method	Calculation time (s)
BCM	3.43
BCM+HC	0.822
BCM+HC+LOD	0.708

BCM: Boundary Condition Map

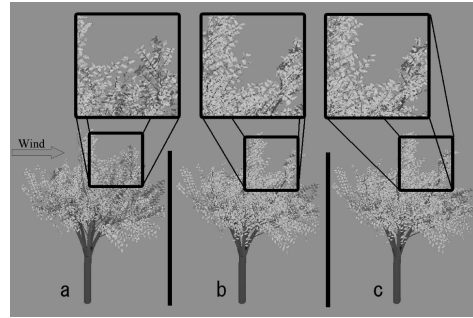
HC: Hierarchical Calculation

LOD: Level of Detail

時間と比較すると、境界条件マップの生成や樹木の揺れの生成に要する時間は微小であることが分かる。

4.3.2 階層的な計算と LOD による高速化

次に、複数の樹木を用いた際の、階層的な計算と LOD による高速化の効果を検証する。図 14 のように、4 本の樹木を均等に配置し、境界条件マップのみを用いた場合、それに加え階層的な計算を行った場合、階層化に加え LOD を適用した場合の 3 種類のケースについて、1 フレームの生成時間を求めた。ただし、境界条件マップを適用する際の格子サイズは $50 \times 20 \times 50$ とし、階層的な計算を行う際の粗い格子サイズは空間全体を $10 \times 4 \times 10$ に分割するものとし、各樹木周囲の格子については、境界条件マップを適用する際の格子サイズに等しい格子で $20 \times 20 \times 20$ に分割するものとする。LOD の実験を行う際は、後方の 2 本の樹木周囲の格子については $15 \times 15 \times 15$ に分割される（粗くする）ように視点位置を設定した。結果を表 4 に示す。



Grid size (a: 5^3 , b: 10^3 , c: 20^3)

図 15 格子サイズによる変形の違い

Fig. 15 Difference of the sway of a tree at each grid size.

この結果から、階層的な計算手法により大幅に計算時間が短縮されていることが分かる。数値計算を行ううえで用いる格子数の合計は 3 つの方法とも大きな差がない。しかし、表 3 を見ても分かるように、格子数が増えるに従い陰解法の収束計算などに時間がかかり、分割数の多い格子では、1 つの格子あたりの計算コストが増大してしまう。一方、階層的な計算を用いた場合には、個々の計算格子のサイズを抑えることができるので、計算が高速化されることが分かる。また、LOD による格子サイズの変更は、樹木ごとに計算量を削減することが可能であるので階層的な計算手法と組み合わせることによって高速化に寄与していることが分かる。

4.3.3 格子サイズによる変形の違い

最後に、格子サイズによる樹木の揺れ方の変化の違いを評価するために、 5^3 、 10^3 、 20^3 の 3 種類の分割数で行った実験結果を示す。その結果、図 15 に示すように、分割数 5^3 の場合のみに枝のしなりが少なくなるという変化が見られた。一方、 10^3 と 20^3 の場合については、わずかな違いはあるものの、ほぼ同様の揺れをするという結果が得られた。 5^3 の分割数では、格子の 1 辺は 1 m となり、約 80 cm の小さな枝やそれに付く長さ 5 cm の葉がすべて 1 つの格子内に含まれてしまうので、均一な抵抗体として扱われる。このため、風の到達しにくい部分で風の回り込みおよび枝のしなりにともなう風速の変化が生じにくくなり、枝のしなりが小さくなった。 10^3 の分割数では、葉の密度変化が大きい部分でも、2~3 回の分割が行われるので、アニメーション上の変化が起こりにくくなっている。この結果から、枝や葉の大きさに対して過剰に分割数を減らした場合には問題が生じるが、分割数を必要以上に細かくしなくてもシミュレーションが可能であることが確認できた。

なお、分割数の増加にともない枝や葉の揺れ(変化)によって境界条件マップの変更が頻繁に行われると、アニメーションの連続性がそのつど失われることが予想されるが、実験結果では、異なる分割数であっても過剰な分割を行わない限り、連続性が失われる度合いには影響を与えないことが確認できた。これは、本研究で用いた SMAC 法による風の物理シミュレーションが、それぞれの時刻で流れ場の定常状態を求めているので、分割数が異なってもほぼ等しい風力を樹木に与えることができるからである。さらに、LOD を用いた際にも、格子サイズが異なってもほぼ等しい揺れ方をすることは、格子を動的に切り換えた場合でも連続性が失われる度合いが少なくなることを表している。

5. おわりに

本研究では、物理シミュレーションに基づく風に揺れる樹木のアニメーション生成を行う際に問題になる計算時間を短縮することを目指し、リアルタイムアニメーションを可能にするための高速化手法を提案し、その効果を検証した。境界条件マップ法と名付けたこの手法の特徴は、枝や葉の均質性に着目し、それらが風に対して及ぼす抵抗を格子単位でモデル化するものであり、格子数の削減を行っても、風に揺れる樹木の様子には大きな変化が見られないことを実験で実証した。そして、一般に行われているその他の高速手法と組み合わせることにより、リアルタイムのアニメーションが実現可能であることを示した。以下に、成果を箇条書きにまとめる。

- (1) 非圧縮性流体における Navier-Stokes の方程式の解法として SMAC 法と呼ばれる数値解法を実装し、上述の特徴を持つ境界条件マップ法を適用することにより、樹木周辺の風の流れを粗い格子で高速に計算し、物理シミュレーションに基づく風に揺れる樹木のアニメーション生成をリアルタイムで行うことを可能にした。
- (2) 風力から枝や葉のしなりを求めるための樹木の力学モデルとしてリンク-ジョイント構造からなる簡易モデルを採用し、シミュレーションに用いる標準的な樹木モデルを定義した。
- (3) 本研究独自に開発した高速化手法である境界条件マップ法に加え、一般的に用いられる階層的な計算手法および LOD の 2 手法を加え、風の計算に要する計算時間をさらに短縮し、高速なアニメーション生成を実現した。
- (4) 各種実験を行い、物理シミュレーションを行うことによってしか表現しえない、樹木周辺にお

ける風の流れの変化によって生じる場所の違いによる枝のしなり方への影響や、障害物による風の回り込みなどを検証するとともに、各種高速化手法による計算時間の短縮の効果を検証した。

なお、本研究では、樹木の力学モデルを簡便にするために標準的な 1 つのモデルしか用意していないが、現在、各種樹木の種の特徴を表現した樹木モデルの研究を並行して行っており、今後は、この研究の成果を本研究に生かすことにより、様々な種類の樹木が風に揺れる様子のアニメーションを行えるようにする予定である。

謝辞 なお、本研究は、平成 14 年度に川崎市で行われたビジネスプラン・コンペかわさき 2002 で奨励賞を受賞した。ここに謝意を表する。

参考文献

- 1) Weber, J. and Penn, J.: Creation and rendering of realistic trees, *Proc. SIGGRAPH '95 Conference*, pp.119-128 (1995).
- 2) Soler, C., Sillion, F., Blaise, F. and Dereffye: A physiological plant growth simulation engine based on accurate radiant energy transfer, *Tech. Rep.*, Vol.4116, INRIA (2001).
- 3) 千葉則茂, 村岡一信, 大川俊一, 三浦 守: CG のための樹木の生長モデル—架空の「植物ホルモン」による自然な樹形の生成, 電子情報通信学会論文誌 D-II, Vol.76, No.8, pp.1722-1734 (1993).
- 4) 金山知俊, 阪田省二郎, 増山 繁: 分枝規則を再現し, 光, ホルモンの影響を考慮した樹木の生長モデル, 電子情報通信学会論文誌 D-II, Vol.79, No.8, pp.1362-1373 (1996).
- 5) 坂口竜己, 大谷 淳: 実写映像に基づいた樹木のモデリング, 電子情報通信学会 MVE 研究会技術報告, pp.25-32 (1997).
- 6) Shlyakhter, I., Rozenoer, M., Dorsey, J. and Teller, S.: Reconstructing 3D Tree Models from Instrumented Photographs, *IEEE Computer Graphics and Applications*, Vol.21, No.3, pp.53-61 (2001).
- 7) Ota, S., Tamura, M., Fujita, K., Muraoka, K., Fujimoto, T. and Chiba, N.: $1/f^\beta$ Noise-Based Real-Time Animation of Trees Swaying in Wind Fields, *Proc. Computer Graphics International*, pp.52-59 (2003).
- 8) Shinya, M. and Fournier, A.: Stochastic motion-motion under the influence of wind, *Computer Graphics Forum*, pp.C119-C128 (1992).
- 9) Giacomo, T.D., Capo, S. and Faure, F.: An Interactive Forest, *Eurographics Workshop on*

Computer Animation and Simulation, pp.65–74 (2001).

- 10) Sakaguchi, T. and Ohya, J.: Modeling and Animation of Botanical Trees for Interactive Virtual Environments, *Symposium on Virtual Reality Software and Technology*, pp.139–146 (1999).
- 11) 金山知俊, 増山 繁: 樹木の揺れのアニメーション, 電子情報通信学会論文誌 D-II, Vol.80, No.7, pp.1843–1851 (1997).
- 12) Perbet, F. and Cani, M.: Animating Prairies in Real-Time, *Proc. Conference on the 2001 Symposium on Interactive 3D Graphics*, pp.103–110 (2001).
- 13) 千葉則茂, 河野 充, 佐藤義人, 村岡一信, 斉藤伸自: 風による樹木の揺らぎ画像生成法の検討, 画像電子学会誌 I, Vol.22, No.5, pp.475–483 (1993).
- 14) Stam, J.: Stable Fluids, *SIGGRAPH 99*, pp.121–128 (1999).
- 15) Foster, N.: Animation and Rendering of Complex Water Surfaces, *SIGGRAPH 02*, pp.736–744 (2002).
- 16) Witting, P.: Computational Fluid Dynamics in a Traditional Animation Environment, *SIGGRAPH 99*, pp.129–136 (1999).
- 17) Rudolf, M.J. and JacekRaczkowski: Modeling the Motion of Dense Smoke in the Wind Field, *Computer Graphics Forum 2000*, Vol.19, No.3, pp.21–29 (2000).
- 18) 杉山 弘, 遠藤 剛, 新井隆景: 流体力学, 森北出版 (1995).
- 19) 梶島岳夫: 乱流の数値シミュレーション, 養賢堂 (1999).
- 20) Amsden, A.A., Harlow and F.H.: The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows, *LA-4370* (1970).
- 21) Perttunen, J., Sievanen, R. and Nikinmaa, E.: LIGNUM: A tree model based on simple structural units, *Annals of Botany*, Vol.77, pp.87–98 (1996).
- 22) Terashima, I., Kimura, K., Sone, K., Noguchi, K., Ishida, A., Uemura, A. and Matsumoto, Y.: Differential analyses of the effects of the light environment on development of deciduous trees: Basic studies for tree growth modeling, *Diversity and Interaction in a Temperate Forest Community*, Vol.158, pp.187–200 (2002).
- 23) Akagi, Y., Sanami, S. and Kitajima, K.: Study on generation of tree shapes with analysis of common features of species, *NICOGRAPH International 2005*, pp.101–106 (2005).



図 16 スギのような樹木
Fig. 16 Trees like a Japanese cedar.



図 17 クスノキのような樹木
Fig. 17 Trees like a camphor tree.

付 録

A.1 他の樹木形状を用いた例

4 章では, 変形結果の違いを明確に示すために同一の樹木を用いた. 付録として他の樹木形状を用いたアニメーション例を示す.

A.1.1 スギのような樹木 (図 16)

A.1.2 クスノキのような樹木 (図 17)

(平成 16 年 9 月 17 日受付)

(平成 17 年 5 月 9 日採録)



赤木 康宏 (学生会員)

平成 13 年東京農工大学工学部電子情報工学科卒業. 平成 15 年同大学大学院修士課程修了. この間, 流体力学を用いた CG アニメーションの研究に従事. 平成 15 年同大学院博士課程入学.



佐波 晶 (正会員)

平成 10 年東京農工大学工学部電子情報工学科卒業. 平成 14 年同大学大学院工学研究科博士課程修了. 同年同大学ベンチャービジネスラボラトリ研究員を経て, 平成 15 年同大工学部情報コミュニケーション工学科助手として機械製品の CAD およびパラメトリックス, 仮想現実の研究に従事, 現在に至る.



北嶋 克寛（正会員）

昭和 46 年東京大学工学部精密機械工学科卒業．昭和 51 年同大学大学院博士課程修了．同年同大学助手，昭和 59 年東京農工大学工学部数理情報工学科（現情報コミュニケーション工学科）助教授，平成 6 年同大学教授，平成 16 年同大学学術研究担当副学長，平成 17 年同大学教授，現在に至る．この間，3D CAD，3D ビジョン，CG，画像理解等の研究に従事．精密工学会，電子情報通信学会，日本機械学会等各会員．
