

ユーザ制御可能な風による樹木の揺れのビジュアルシミュレーション

朱 暁宇¹⁾, Abdul Hadi Bin Mohdnazir²⁾, 伊藤弘樹³⁾, 菊池 司⁴⁾

^{1) 2)} 拓殖大学大学院, ^{3) 4)} 拓殖大学

{¹⁾ z9m404, ²⁾ z1m410}@st.takushoku-u.ac.jp, {³⁾ k-itoh, ⁴⁾ tkikuchi}@id.takushoku-u.ac.jp

Visual Simulation of Trees Swaying by Controllable Wind Field

Xiaoyu Zhu¹⁾, Abdul Hadi Bin Mohdnazir²⁾, Kouki Itoh³⁾, Tsukasa Kikuchi⁴⁾

^{1) 2)} Graduate School of Engineering, Takushoku University, ^{3) 4)} Takushoku University

{¹⁾ z9m404, ²⁾ z1m410}@st.takushoku-u.ac.jp, {³⁾ k-itoh, ⁴⁾ tkikuchi}@id.takushoku-u.ac.jp

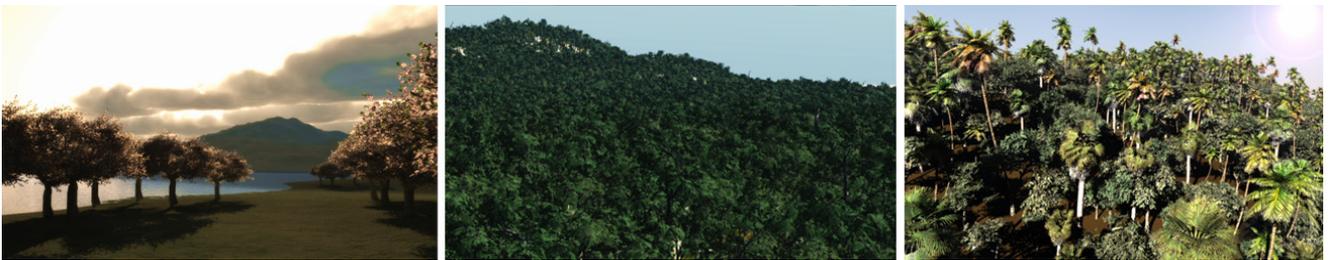


図 1. 大規模な森林景観の例. 本論文ではこのような景観映像において, 樹木が小さくランダムに揺れる小揺れとユーザが生成した風の影響による大揺れを表現する手法を提案する.

アブストラクト

本論文では, 大規模な森林景観をコンピュータグラフィクス (CG) で再現する際に重要な要素のひとつとなる「風による樹木の揺れ」を, ユーザが制御できるように生成する手法を提案する.

本手法では, 樹木モデル 1 本分を包含するようにバウンディングボックスを配置し, 葉 1 枚分に相当するサイズでボクセルに分割する. そして, ボクセルを構成する 4 頂点にパーティクル (ボクセルパーティクル) を配置し, ボクセルパーティクルを $1/f$ ノイズによって運動させることによってバウンディングボックスを変形させ, 樹木のランダムな揺れ (以後, 小揺れ) を再現する. さらに, シミュレーション空間に樹木データを複数本配置し, ユーザがスプラインを描画することによりスプラインに沿った風を生成する手法を提案する. この風とボクセルパーティクルとの衝突判定を行うことによって, 風の影響による樹木の揺れ (以後, 大揺れ) を再現する.

本手法により, 大規模な森林景観において樹木が小さくランダムに揺れる小揺れと, ユーザが生成した風の影響による大揺れを表現することが可能となり, ユーザが思い描いたような「風の通り道」が見える, 印象的なシーンを生成することができる.

Keywords: ビジュアルシミュレーション, 大規模森林景観, 樹木の揺れ, 制御可能な風, バウンディングボックス

Abstract

In this paper, we propose a method to create animation with trees swaying in the wind that a user can control. This method enables the user to make animations of large-scale forest scenes by Computer Graphics (CG).

In our method, we arrange a bounding box to include one tree model, and divide it in voxels at a size equivalent of one leaf. We represent random swaying (slight swaying) of the tree by transforming the voxels by $1/f$ noise. Furthermore, we propose a method for arranging data on several trees in the simulation space and to generate wind along a spline drawn by the user. We represent larger swaying of the trees by the wind by collision between the wind and the voxels.

By this technique, animations of trees swaying in a large-scale forest scene can be produced by “wind” that the user has generated.

Keywords: Visual Simulation, Large-scale Forest Scene, Tree Swaying, Controllable Wind, Bounding Box

1. はじめに

本論文では、屋外景観をCGで再現する場合に重要な要素となる「樹木」を表現対象とし、複数本の樹木による大規模な森林景観(図1参照)において、「風による樹木の揺れ」を流体運動方程式(Navier-Stokes, 以後、NS方程式)を用いずにユーザが制御できるように生成するビジュアルシミュレーション法を提案する。本手法では、樹木モデル1本分を包含するようにバウンディングボックスを配置し、葉1枚分に相当するサイズでボクセルに分割する。そして、ボクセルを構成する4頂点にパーティクル(ボクセルパーティクル)を配置し、ボクセルパーティクルを1fノイズによって運動させることによってバウンディングボックスを変形させ、樹木のランダムな揺れ(以後、小揺れ)を再現する。さらに、シミュレーション空間に樹木データを複数本配置し、ユーザがスプラインを描画することによりスプラインに沿った風を生成する手法を提案する。この風とボクセルパーティクルとの衝突判定を行うことによって、風の影響による樹木の揺れ(以後、大揺れ)を再現する。すなわち、ランダムな小揺れとユーザが制御可能な大揺れの2つを組み合わせることによって、ユーザが思い描いた「風の通り道」が見えるような印象的なシーンを生成することが可能となる。

CGでの樹木の形状モデリングに関しては、文献[1-3]のみならず、これまでに数多くの手法が提案されている。しかしながら、これらの研究例では樹木の形状だけをモデリングすることに主眼が置かれている。

樹木が風によって揺れ動く運動をシミュレーションした研究例としては、[4-7]らが挙げられる。これらの研究例では、樹木の枝が風によって運動する様子の再現を、個々の枝を運動させるための力を考慮し物理シミュレーションを行うことによって実現している。しかしながら、物理シミュレーションによるアプローチは、風フィールドと枝の間の相互作用力による影響を忠実に再現できるという利点がある反面、計算コストがかかるという問題点もある。さらに、これらの研究例では、枝に付いている葉1枚1枚の運動までは再現されていない。現実の樹木の風による揺らめきを観察していると、枝が風によって運動する様子よりも、葉1枚1枚が複雑に揺らめく運動のほうが視覚的な特徴としては重要であるように思われるが、これらの研究例ではそれが示されていない。

樹木の風による運動をシミュレーションした研究例には他にも、J.DienerらによるGPUを利用した研究例[8]がある。この研究では、1本の樹木を有限個の制御点に分割し、制御点を振動関数によって運動させるシミュレーションをGPUを用いて計算することによって、大規模な森林景観における風の影響をリアルタイムで再現する手法を提案している。しかしながら、J.Dienerらの手法では、風は均一な空間分布であるという仮定に基づいており、風の通り道をユーザが制御できることは示されていない。また、葉は枝の運動に追従するだけで、葉自身が風の影響で運動する様子はシミュレーションされていない。

J.Dienerらと同様に、GPUを利用した物理シミュレーションを行なっている研究例として、R.Habelらの手法[9]が挙げられ

る。R.Habelらの手法では、樹木の運動を再現するために枝モデルを分割する必要がない非線形の変形モデルを提案し、樹木と風のインタラクションは、樹木全体を周波数空間で包含することによって乱流による枝の応答関数を事前に計算しておき、2次元のモーシオンテキストチャとして枝の運動に利用することにより、風の影響をリアルタイムで計算する。また、R.Habelらの手法では葉の運動も考慮され、自然な風の影響を再現できる様子が示されているが、大規模な森林景観に適用できることは示されておらず、さらに、風の通り道をユーザが制御できることは示されていない。

S.Otaらによる研究例[10]では、運動方程式による物理シミュレーションを行うことなく、葉と枝の自然な運動をリアルタイムで生成する手法が提案されている。S.Otaらの手法は、葉の運動を1fノイズによって生成し、枝の運動は単純なバネモデルによって生成するという複合型のアプローチとなっている。本手法によって、1fノイズのパラメータを変化させることにより、風フィールドにおける樹木の揺れ方をユーザがリアルタイムに試行錯誤できることが示されているが、大規模な森林景観に適用できることは示されていない。

そこで本論文では、大規模な森林景観における風フィールドによる樹木の揺れを、ユーザが制御できるように生成する手法を提案する。

本論文は以下のように構成される。第2章では、本論文で扱う樹木モデルと樹木モデルの運動を制御するバウンディングボックスに関して述べる。第3章では、ボクセルパーティクルを1fノイズで運動させることによってバウンディングボックスを変形させ、樹木モデルの葉や枝が細かく運動する“小揺れ”を再現する手法に関して述べる。第4章では、ユーザ制御可能な風の生成法と、風とボクセルパーティクルの衝突による大揺れの表現に関して述べる。第5章では、本論文で提案する手法によるシミュレーション結果を示し、シミュレーション結果に関しての議論を行う。最後に、第6章で本論文のまとめと今後の課題に関して述べる。

2. 樹木モデルとバウンディングボックス

本章では、本論文で扱う樹木モデルと、樹木モデルの運動を制御するためのバウンディングボックスに関して説明する。

2.1 樹木モデルの概要

本論文で提案する手法は「風による樹木の運動」に主眼を置いているため、樹木モデルの生成方法に関してはこれまでに提案されているようなアルゴリズムによる自動生成法をインプリメントして使用するのではなく、市販されている樹木データをテンプレートとし、樹木写真などを観察しながらクリエイターが手作業で編集を加えるという手法を採用した。

そのようにして生成された樹木のポリゴンモデルにおいて、樹木の葉と枝はそれぞれ違う属性のポリゴンとして扱うものとする。すなわち、葉ポリゴンと枝ポリゴンのように判別が可能ないようにポリゴン個々に識別フラグを付ける。葉ポリゴンと枝

ポリゴンが判別可能であるということが、本手法を用いるための必要条件である。

葉ポリゴンの各頂点は、質点バネモデル (Mass-Spring Model) [11]を用いて連結する。質点バネモデルとは、図 2 に示すように質点が質量を持たないバネによってつながれたもので、構造バネ、せん断バネ、および屈曲バネの3種類の役割を果たすバネによって構成される。構造バネは縦方向と横方向の基本構造を作るためのバネであり、せん断バネは質点間が重力などによって斜め方向に極端にズレることを防ぐためのバネである。また、屈曲バネは曲げに対する回転復元力を実現する。屈曲バネにより、横方向、および縦方向の構造バネに接続されている1つ離れた質点同士は互いに反発しあう力が発生し、葉の形状を保つ力を強めることが可能となる。

質点*i*に対して働く力の総和 \mathbf{F}_i は、以下の運動方程式 (1) により定式化される。

$$\mathbf{F}_i = \sum_j \left\{ k_{ij} \left(1 - \frac{L_{ij}}{|\mathbf{r}_{ij}|} \right) \mathbf{r}_{ij} + D_{ij} \mathbf{v}_{ij} \right\} \quad (1)$$

ここで、 k_{ij} , L_{ij} , D_{ij} はそれぞれ、バネ係数、バネ自然長、ダンパ係数を表し、 \mathbf{r}_{ij} , \mathbf{v}_{ij} は質点*j* に対する質点*i* の相対位置ベクトル、および相対速度ベクトルを表す。

本手法で用いる葉ポリゴンにおいて、枝との接点に近いポリゴン頂点を支点にして葉全体が運動しているようにしたいため、式 (1) におけるバネ係数 k_{ij} とダンパ係数 D_{ij} に重み付けを行なう。すなわち、枝との接点に近い葉ポリゴンの頂点 (質点) ほど、バネ係数、およびダンパ係数が大きいものとし、枝から離れる (葉の先端のほうに向かう) にしたがって、バネ係数、およびダンパ係数は線形的に減少するものとする (図 3 参照)。

枝ポリゴンに関しては、第 3 章、および第 4 章で述べる枝の運動を制御するため、枝ポリゴン内部にボーンを設定する (図 4 参照)。ボーンは枝が分かれる節ごとに設定し、樹木の幹から先端の小枝に向かって親子関係を定義する (枝の先端部分が子になる)。枝の運動は、エンドエフェクタである子ボーンの先端が運動することによって、その運動量が親ボーンへ伝播していくインバースキネマティクス (IK) を用いることで実現する。

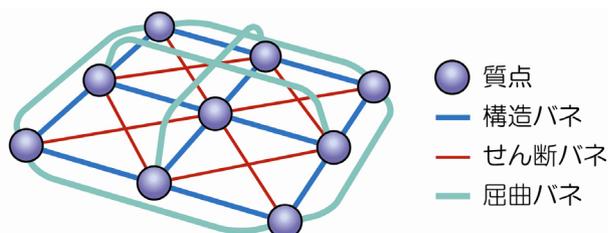


図 2. 3 種類のバネによる質点バネモデル

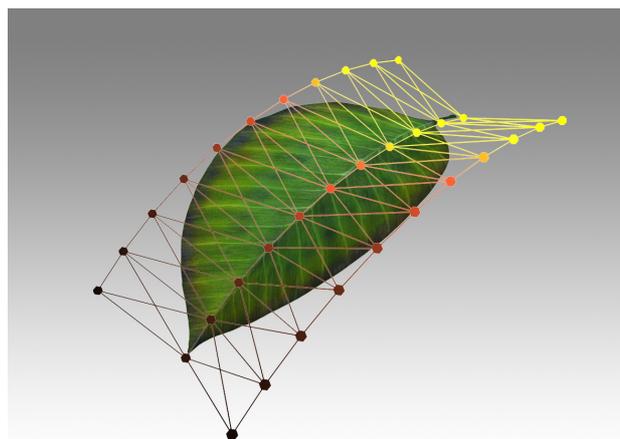


図 3. 構造バネ、およびせん断バネとバネ係数重み付けの例。黄色から黒へのグラデーションにおいて、明るい色の部分がバネ係数とダンパ係数が大きい。

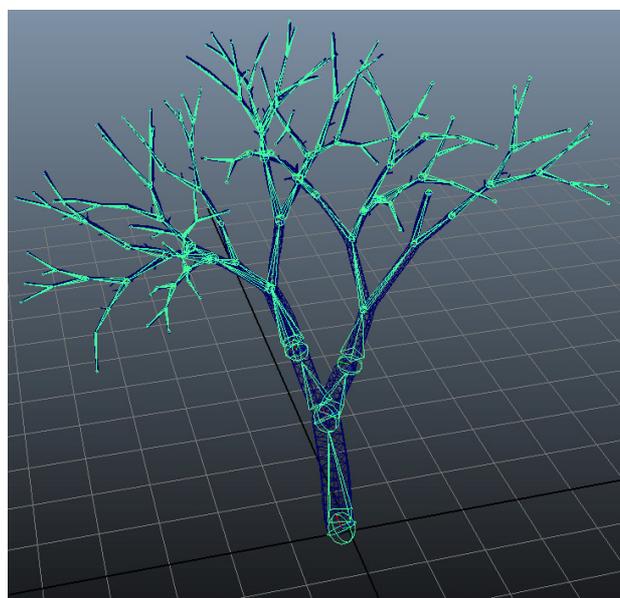


図 4. 枝ポリゴン内のボーンの例。樹木データ (ポリゴン) は紺色のワイヤーフレームで表示し、ボーンは明るい緑色でのアウトライン表示としている。

2.2 樹木モデルを包含するバウンディングボックスの配置

前述の樹木モデル1本分を包含するようにバウンディングボックスを配置し、葉1枚分に相当する大きさにボクセルに分割する (図 5 参照)。樹木モデルの葉ポリゴンの頂点と枝ポリゴンに内蔵されたボーンのエンドエフェクタを、包含するボクセルに登録し管理する (図 6 参照)。

さらに、各ボクセルの頂点にパーティクルを配置し、パーティクルの3次元空間内の座標と各ボクセル頂点の座標はリンクしているものとする。これをボクセルパーティクルと呼ぶ (図 7 参照)。すなわち、ボクセルパーティクルが運動することにより、各ボクセル頂点も連動して運動し、バウンディングボックスが変形することとなる。

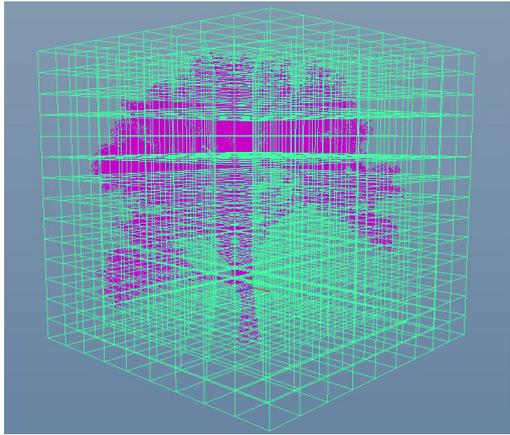


図5. 樹木1本分のバウンディングボックス. ボクセル分割を行なった様子を明るい緑色で表示している (図中のボクセルは, 図の見やすさの観点から大きく分割している).

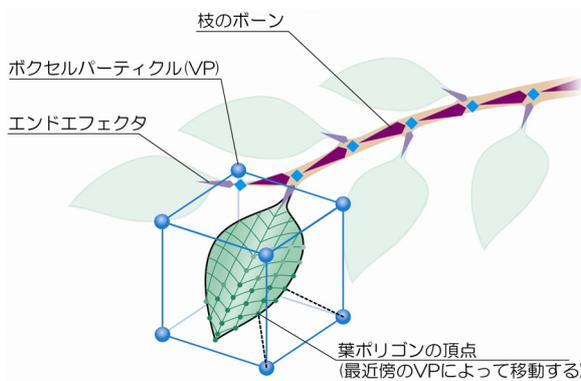


図6. 葉ポリゴン, エンドエフェクタ, およびボクセルの様子

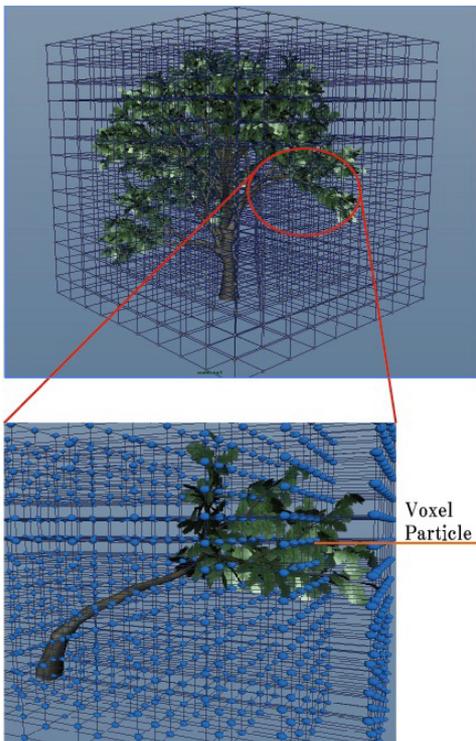


図7. 各ボクセル頂点に割り当てられたボクセルパーティクル (図中のボクセルは, 図の見やすさの観点から大きく分割している)

2.3 樹木モデルのボクセルへの登録と運動モデル

前節でも述べたように, 各ボクセルには樹木モデルの葉ポリゴンの頂点, および枝ポリゴンに内蔵されたボーンのエンドエフェクタが登録されている.

葉ポリゴンの頂点, およびエンドエフェクタは, 以下の 4 ステップの計算により運動する.

【ステップ 1】:

葉ポリゴンの頂点, およびエンドエフェクタは, 登録されているボクセルの最近傍ボクセルパーティクルの加速度ベクトルの影響を受け, 次の式 (2) の力によって移動する.

$$\mathbf{F}_i = \frac{\mathbf{a}_{vp}}{R_{ivp}}$$

$$\mathbf{a}_{vp} = \frac{d\mathbf{v}_{vp}}{dt} \quad (2)$$

ここで, \mathbf{F}_i は葉ポリゴンの頂点 i , もしくはエンドエフェクタ i が受ける力 (式 (1) と同義), \mathbf{a}_{vp} は最近傍ボクセルパーティクルの加速度ベクトルで, ボクセルパーティクルの速度ベクトル \mathbf{v}_{vp} の単位時間 t についての微分によって求められる. R_{ivp} は葉ポリゴンの頂点 i , もしくはエンドエフェクタ i と最近傍ボクセルパーティクル間の距離である.

【ステップ 2】:

葉ポリゴンの頂点に関して, 葉 1 枚を構成している全頂点に関して, 上記ステップ 1 を計算する. その後, 最も大きな力 \mathbf{F}_i を受ける頂点を検索し, 他の頂点に関しては力 \mathbf{F}_i をリセットする. なお, 葉ポリゴンの頂点とボクセルパーティクルが接触するとゼロ除算エラーとなるが, その場合, \mathbf{F}_i はゼロであるものとし, 次のステップ 3 によって他の葉ポリゴンの頂点の影響を受けて葉全体が運動することとなるため, 動作上の問題はない.

【ステップ 3】:

葉ポリゴンの頂点において, ステップ 2 で力 \mathbf{F}_i をリセットされた頂点は, 式 (1) の質点パネモデルによって移動する. すなわち, 葉 1 枚分全てのポリゴン頂点は必ずしも同一のボクセルに登録されるわけではないが, 葉の運動を決定するのは最も大きな力を受けた 1 頂点のみであり, 他の頂点は質点パネモデルによってこの 1 頂点に追従する形で葉全体が運動することになる.

【ステップ 4】:

枝のエンドエフェクタにおいては, 式 (2) による最近傍ボクセルパーティクルからの力 \mathbf{F}_i によって運動する.

上記の運動モデルにより, 葉ポリゴンの形状と樹木全体の形状が大幅に破綻することなく, ボクセルパーティクルの運動に連動するように樹木, および葉を運動させることができる (図8参照).

なお, 本論文で提案する手法は大規模な森林景観における風

の影響を受けて揺らめく樹木のビジュアルシミュレーションを対象にしているため、葉や枝同士の衝突判定は行っていない。したがって、例えば同一ポリゴン内に複数の葉や枝が登録されており、運動の際にそれらが同一方向に運動して衝突していたとしても、シーン全体での運動を考慮した場合にはそれらは無視できるものとして扱う。

さらに、上記のアルゴリズムで枝のエンドエフェクタが運動すると、運動量によっては枝が伸縮するという現象が発生するが、これもシーン全体での運動を考慮した場合には無視できるものとして扱い、本論文ではこの状況を回避するための処理を行わないものとした。

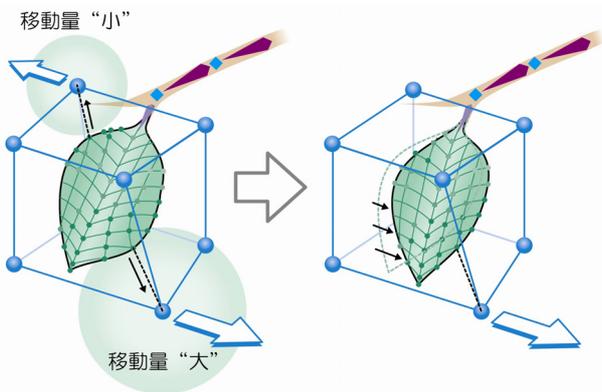


図8. ボクセルパーティクルによる葉と枝の運動

3. バウンディングボックスの変形による小揺れの表現

本章では、ボクセルパーティクルを $1/f$ ノイズで運動させることによってバウンディングボックスを変形させ、樹木モデルの葉や枝が細かく運動する“小揺れ”を再現する手法に関して述べる。

本論文では、小揺れを再現する力 F_{small} として以下の式 (3) ような力を定義し、オイラー法によりボクセルパーティクルの運動シミュレーションを行う。

$$F_{small} = f_n + Pf_m \quad (3)$$

ここで、 f_n はボクセルパーティクルが $1/f$ ノイズの影響を受けて運動する際に働く力であり、 Pf_m は各ボクセルの体積を近似的に保存するためのボクセルパーティクル間力[12]である。

これらの力と、樹木モデルが変形する際の枝ポリゴンの“潰れ”を防ぐための SSD 法 (Skeletal Subspace Deformation) [13]に関して、以下で説明する。

3.1 $1/f$ ノイズの概要

関数 $X(t)$ のフーリエ変換の 2 乗平均をスペクトル濃度 $S(f)$ とすると、 $1/f$ ノイズとは、スペクトル濃度が $1/f$ と比例しているノイズである[14]。ここで、 t は時間、 f は周波数を表している。また、 β の値によってノイズの変動を制御できるよ

うにしたものを、 $1/f^\beta$ ノイズと言う [10]。

$1/f$ ノイズは、様々な自然現象から観測されることが報告されている[15, 16]。本論文における樹木の枝葉を運動させる力として $1/f$ ノイズを採用した理由も、現実世界の自然の風からも $1/f$ ノイズが観察されるという事実に基づいている。

$1/f^\beta$ ノイズの生成法に関しては、本研究では参考文献[10]に詳細が述べられている手法を利用しており、本論文で提案する手法の主要部分ではないため、説明を割愛する。詳細は、参考文献[10]を参照されたい。

本研究では、参考文献[10]の手法により 3 つの $1/f^\beta$ ノイズを同時に生成し、それぞれをボクセルパーティクルの x, y, z 座標に働くノイズ、すなわち f_n とする。 f_n は以下の式 (4) のように表される。

$$f_n(x, y, z) = \left(\frac{1}{f_1^\beta}, \frac{1}{f_2^\beta}, \frac{1}{f_3^\beta} \right) \quad (4)$$

ここで、 $\frac{1}{f_1^\beta}, \frac{1}{f_2^\beta}, \frac{1}{f_3^\beta}$ はそれぞれ異なる $1/f^\beta$ ノイズであることを表している。

そして、ボクセルパーティクルの各座標に対して $f_n(x, y, z)$ を割り当て、運動シミュレーションを行う。

なお、 $f_n(x, y, z)$ はシミュレーション時間ステップとノイズの時系列変化を同期させ、シミュレーションステップ毎に更新を行なう。すなわち、 $f_n(x, y, z)$ によってボクセルパーティクルがシミュレーションステップ毎に微細に運動することとなり、それに連動してバウンディングボックスが変形 (運動) することによって、バウンディングボックス内に登録された樹木データが 2.4 節で述べた運動モデルにしたがって変形 (運動) することとなる。

3.2 ボクセルパーティクル間力

前節の $1/f^\beta$ ノイズを利用した運動の生成法では、ボクセルパーティクルの急激な運動によってボクセルの体積が保たれない場合がある、すなわちボクセルが潰れてしまったり、隣接するボクセルパーティクルと交差してしまうという問題点がある。

そこで我々は、各ボクセルの体積を近似的に保存するために、ボクセルパーティクル間に働く斥力の存在を定義する。また、ボクセルパーティクル間が f_n の影響により離れすぎないように、引力も定義する。

本モデルでは、参考文献[12]で提案されている以下の相互作用力を利用し、ボクセルパーティクル間力 Pf_m とする。

- (a) 安定状態では相互作用力は働かない
- (b) ある一定距離より小さいときは斥力が働く
- (c) ある一定距離より大きいときは引力が働く

なお、本論文では図 9 に示すグラフに基づいた相互作用力を用いるものとし、ボクセルパーティクル間の距離は、ボクセルの初期状態 (樹木モデル 1 本を包含するようにバウンディングボックスを配置し、葉 1 枚分の大きさでバウンディングボック

スをボクセルに分割した状態)で定義されたボクセルパーティクル間の距離が、図9の横軸における「1」となるように設定するものとする。

ボクセルパーティクル間力により、バウンディングボックス内部のボクセルパーティクルは安定状態に近い状態が保たれるようになる。すなわち、樹木の内側の葉は運動しにくくなる。

また、バウンディングボックスの外側に位置するボクセルパーティクルにおいても、不安定な運動や急激な運動を示すことがなくなるため、樹木データのポリゴンが大幅に破綻するような様子が見られなくなるという利点がある(図10参照)。

なお、枝の運動に関しては2.4節で述べたように、エンドエフェクタが最近傍のボクセルパーティクルの影響を受けて運動するという単純なものであるため、風の影響による揺れ戻し(右に揺れた反動で次は左に揺れるような現象)は考慮していないが、ボクセルパーティクル間力によってバウンディングボックスが元の形に復元しようと運動する際に枝のエンドエフェクタが追従して運動することによって、揺れ戻しの現象が再現可能である。

3.3 枝ポリゴンにおけるSSD法

枝ポリゴンに内蔵されたボーンのエンドエフェクタを2.4節で述べた運動モデルにしたがって運動させ、親子関係で階層付けされている親ボーンへIKにより運動が伝播していくことで、枝の運動を実現する。

ボーンが運動することにより、枝ポリゴンが変形して枝の運動を表現するわけであるが、その際にはSSD法(Skeletal Subspace Deformation) [13]を用いる。

SSD法は、キャラクターアニメーションの分野において、キャラクターの関節を任意の形に変形するために開発され、物体のスキン(頂点座標)をボーンで近似し、ボーンの変形から線形ブレンドによりスキンを変形させるという手法である。

SSD法は、解剖学的に正しくない計算方法であり、腱の隆起や皮下組織の圧縮膨張などが表現できないなどの問題点が指摘されているが、本研究における表現対象である枝の運動では、キャラクターの関節のような急激な運動や、90度以上の角度で曲がる関節を表現する必要はないため、SSD法を用いることとした。SSD法の詳細に関する記述は割愛するので、参考文献[13]を参照されたい。

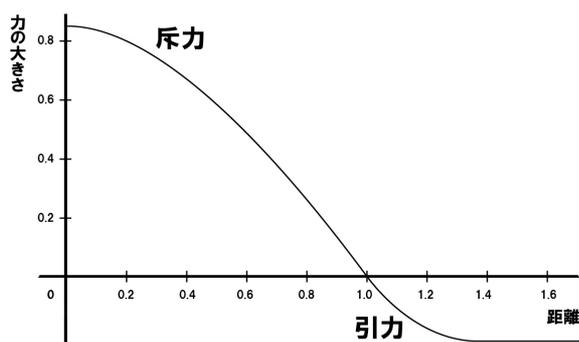


図9. 相互作用力

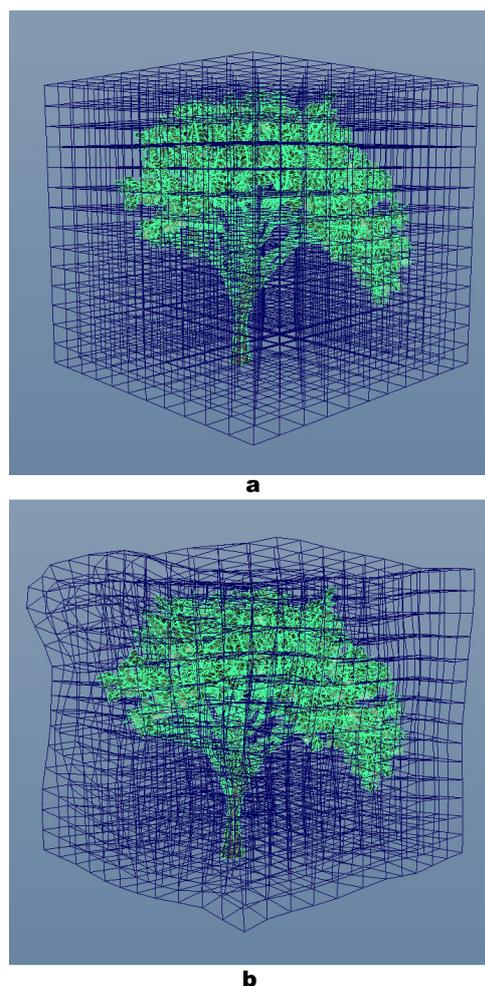


図10. バウンディングボックスの変形による樹木モデルの変形。aは初期状態、bはボクセルパーティクルの運動シミュレーションによるバウンディングボックスと樹木モデルの変形例。

4. ユーザ制御可能な風による大揺れの表現

本章では、ユーザ制御可能な風の生成法と、風パーティクルとボクセルパーティクルの衝突による大揺れの表現に関して述べる。

4.1 スプライン描画による風パーティクルの生成

シミュレーション空間内に風フィールド(流体フィールド)を生成する手法は、NS方程式を用いて流体シミュレーションを行う手法[17]や、ノイズによる手法[10][18]が提案されているが、本論文では、風に見立てたパーティクルを用いることで、ユーザが制御可能な風フィールドを実現する手法を提案する。風に見立てたパーティクルのことを、我々は風パーティクルと呼ぶ。

本手法では、ユーザがシミュレーション空間上に任意に描いたスプラインを風の通り道と定義し、スプライン上に風パーティクルを発生させる(図11参照)。風パーティクルの発生源は、ユーザが描いたスプラインの起点とし、風パーティクルの初速度は任意とする。速度ベクトルの方向は、スプラインの方向に

ランダムな角度誤差を加えた方向とする。

風パーティクルは、ユーザが決定可能な任意の回数によって分裂して発生する。第1段階目の発生(図11中の青パーティクル)は、風パーティクルがスプラインの起点から発生した最初の風パーティクルであり、そこから任意のシミュレーションステップを経過すると、第1段階の風パーティクルから分裂する形で第2段階目(図11中の黄色パーティクル)が発生する。同様に、第2段階目のパーティクル発生から任意のシミュレーションステップを経過すると、第2段階目のパーティクルから分裂する形で第3段階目(図11中の赤パーティクル)が発生する。この分裂発生を、任意回数繰り返す。各段階のパーティクルは、発生源となる1段階前のパーティクルの速度ベクトルにランダムな角度差を与えた上で、発生時の速度値を継承して発生するものとする(図12参照)。

この数段階のパーティクルによって、第1段階目の風パーティクルは風向きを再現し、第2段階目以降の風パーティクルは風の幅、および風の空間的複雑性(乱流成分)を再現する。すなわち、風パーティクルがボクセルパーティクルと衝突することによってバウンディングボックスが変形し、樹木の大揺れを再現するわけだが(詳細は次節で述べる)、風パーティクルを複数回分裂発生させることにより、風の強さ、および風の空間的複雑性(乱流成分)をシミュレートすることが可能となる。

風の強さは、風パーティクルの速度ベクトルと発生するパーティクル数によって制御可能である。速度ベクトルの方向は、与えられた初期段階の方向を保ち、速度値は初速度が単純に一次式で減衰するものとする(減衰比は任意)。風パーティクルの発生、および分離数は、ユーザが任意で決定する。また、各段階の風パーティクルは、発生から任意のステップ数を経過した後、消滅するものとしている。

なお、図11と図12は風パーティクルの分裂回数を「3」と設定した場合の例である。

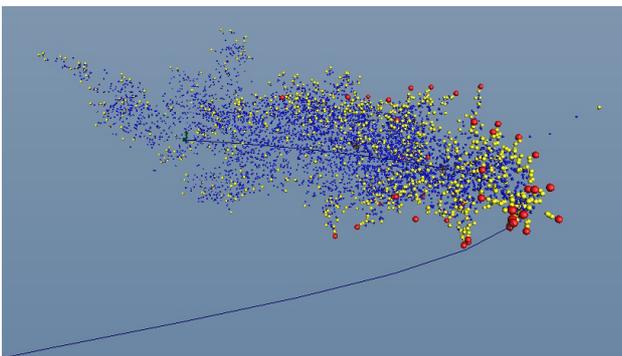


図11. ユーザ描画のスプラインと3段階発生した風パーティクル。図中の黒いラインがユーザ描画のスプラインであり、そのラインに沿って発生しているのが風パーティクル。青の風パーティクルが第1段階目、黄色の風パーティクルが第2段階目、および赤の風パーティクルが第3段階目である。

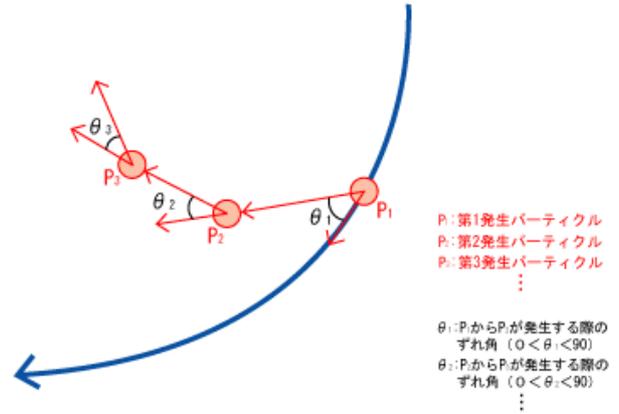


図12. 各段階の風パーティクルの概略図

4.2 風パーティクルとボクセルパーティクルの衝突による大揺れの生成

前節で述べた風パーティクルとボクセルパーティクルとの衝突を計算し、その際の衝突力によってボクセルパーティクルが運動することで、ユーザが描いた風による樹木の揺れ(大揺れ)を実現する。

大揺れを再現するためのボクセルパーティクルが風パーティクルから受ける力 \mathbf{F}_{big} は、以下の式(5)で計算される。

$$\mathbf{F}_{big} = \sum_p \frac{\Delta \mathbf{v}_p}{\Delta t} \quad (5)$$

ここで、 p は計算対象となっているボクセルパーティクルに衝突する風パーティクルの数、 $\Delta \mathbf{v}_p$ は風パーティクルの速度ベクトルの変化量、および Δt はシミュレーションの単位時間の変化量である。

風パーティクルとボクセルパーティクルの衝突判定計算は、以下のように行う。

(1) 樹木1本分を包含しているバウンディングボックスを分割しているボクセルと同じ大きさで分割された衝突計算空間を、ユーザが描いたスプライン1本を包含するように定義する(図13参照)。具体的には、衝突計算空間の最小座標値を(Coll_minx, Coll_miny, Coll_minz), 最大座標値を(Coll_maxx, Coll_maxy, Coll_maxz), ユーザが描いたスプラインの最小座標値を(Spline_minx, Spline_miny, Spline_minz), 最大座標値を(Spline_maxx, Spline_maxy, Spline_maxz)とすると、衝突計算空間は以下の式(6)のように定義する。ここで、 a と b は衝突計算空間の大きさを制御する任意定数であり、 H_b は樹木1本分を包含しているバウンディングボックスの高さの最大値(衝突計算空間内に含まれるバウンディングボックス中)である。なお、衝突計算空間から外れた風パーティクルに関しては、衝突判定は行わないものとする。

$$\begin{aligned}
 Coll_min x &= Spline_min x - \frac{(Spline_max x - Spline_min x)}{a} \\
 Coll_max x &= Spline_max x + \frac{(Spline_max x - Spline_min x)}{a} \\
 Coll_min y &= Spline_min y - b \times H_b \\
 Coll_max y &= Spline_max y + b \times H_b \\
 Coll_min z &= Spline_min z - \frac{(Spline_max z - Spline_min z)}{a} \\
 Coll_max z &= Spline_max z + \frac{(Spline_max z - Spline_min z)}{a}
 \end{aligned}
 \tag{6}$$

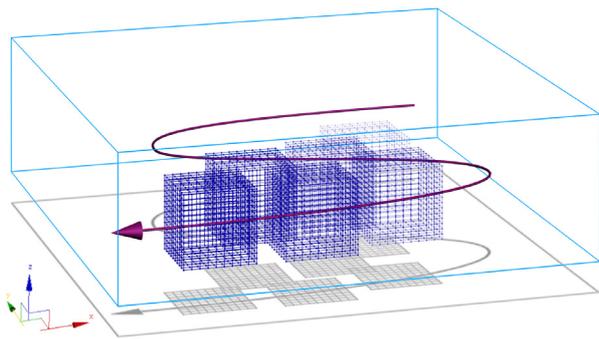


図13 衝突計算空間の概略図. 濃い青の立方体(グリッド表示)は、樹木のバウンディングボックスを表し、薄い青のラインは衝突計算空間を表す。ただし、図の見やすさの観点から、衝突計算空間に関してはグリッドを非表示とした。

(2) シミュレーションステップ毎に、衝突計算空間の各ボクセルに包含される風パーティクルとボクセルパーティクルを登録する。

(3) 衝突計算空間の各ボクセルに登録されている風パーティクルとボクセルパーティクルにおいて、パーティクルの中心座標と半径、およびパーティクル間の距離から衝突判定を行い、風パーティクルとの衝突が検知されたボクセルパーティクルに関して、式(5)によって風パーティクルから受ける力を計算する。風パーティクルに関しては、ボクセルパーティクルとの衝突が検知された場合でも、速度ベクトルの変更(例えば、衝突による屈折や反射など)は行わないものとし、速度値は一次式で減衰するのみである。

(4) 風パーティクルとの衝突が検知されたバウンディングボックスにおいても、第3章で述べた小揺れのための力 F_{small} の計算を行なう。すなわち、ボクセルパーティクルに働く力 F_{vp} の総和は以下の式(7)のようになる。

$$\mathbf{F}_{vp} = \mathbf{F}_{big} + \mathbf{F}_{small} \tag{7}$$

さらに風パーティクルの移動を行い、(2)からの処理をシミュレーションステップ数だけ繰り返す。

図14に風パーティクルとボクセルパーティクルの衝突による、大揺れのシミュレーション例を示す。図14において、風パーティクルの分裂回数は「3」と設定している。

なお、ユーザが描けるスプラインの数、すなわち発生させられる風の数に制限はないため、ユーザは自由に複雑な風フィールドを生成することができる。

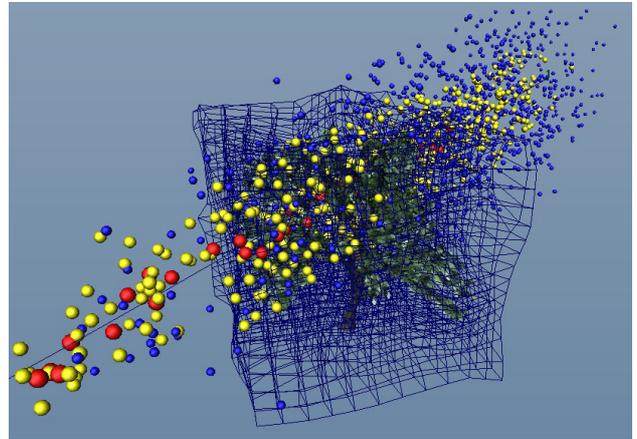


図14. 風パーティクルによる大揺れのシミュレーション

5. シミュレーション結果と検証

本章では、本論文で提案する小揺れと大揺れを組み合わせた樹木の揺れのシミュレーション手法によって生成された画像例を示し、シミュレーション結果に関して検証を行なう。

5.1 シミュレーション環境

シミュレーションに使用した環境は、以下の通りである。

- 使用計算機 : HP WORKSTATION Z600,
- CPU : Xeon E5504@2.00GHz (4 コア)
- メモリ : 8 GB
- 使用 OS : Windows 7 (64bit)
- Graphics card : NVIDIA Quadro FX 1800
- 樹木1本に対する枝のポリゴン数 : 14640
葉のポリゴン数 : 17304
- ボクセルへの分割数 : 100 x 100 x 100
- シーン内の樹木数 : 200 本
- $1/f^\beta$ ノイズの β : 2.5
- 葉ポリゴンのバネ係数 k_{ij} : 0.01 ~ 1
- 葉ポリゴンのダンパ係数 D_{ij} : 0.01 ~ 1
- ユーザ描画のスプライン : 1 本
- 風パーティクル数 : 第1段階目は1フレームにつき200個発生, 第2段階目は1フレームにつき第1段階目のパーティクルから2個分離, 第3段階目は1フレームにつき第2段階目のパーティクルから1個分離。
- 風パーティクルの初速度 : 0.25 (さらに乱数を加算)
- 衝突計算空間の大きさを定義する任意定数 : $a=2$, $b=1$

なお、本シミュレーションは Autodesk Maya2011 の MEL Script を使用して、Maya2011 上で動作するプラグインとして実装したものであり、レンダリングに関しては V-Ray を使用している。

5.2 シミュレーション結果, および検証

シミュレーション結果を, 図 15 (最終頁に掲載) に示す. 本シミュレーション結果は, 我々が先行研究[19]において提示したシミュレーション結果に新たなシーンを加えて, 樹木 1 本のみのシーンから大規模な森林景観に対して, 本論文で提案した手法の有効性を検証したものとなっている.

本シミュレーション結果は, 総フレーム数 300 でシミュレーション時間が 8 分 23 秒であった. レンダリング時間に関しては, 本論文で提案する手法の主要部分ではなく, 光源の数や樹木の本数, およびカメラワークなどのシーン設定にもよるが, 上記シミュレーション時間を計測したシーンデータ (シーン内の樹木本数 200 本, 総フレーム数 300) において光源を 8 個使用し, 約 40 分であった.

図 15 に示すシミュレーション結果より, ボクセルパーティクルの運動に連動して樹木の葉と枝が運動することにより, 計算コスト, およびユーザの作業量を軽減しながら, ユーザの思い描いた風の通り道に沿って樹木が揺れ動くアニメーションが生成可能であることが確認された.

また, 3.2 節で述べた枝の揺れ戻しに関しては, バウンディングボックスが復元しようとする運動に追従する形で再現可能であり, 揺れ戻しの力による慣性力による振動現象は, ボクセルパーティクルが安定状態になるまでの過程で振動をすることがあるが, その影響によって枝のエンドエフェクタも振動するため, 結果として慣性力による振動現象も再現可能である.

しかしながら, 2.4 節でも述べたように, 画面をクローズアップにすると, 運動の途中で隣接する樹木同士のバウンディングボックスが重なり合うこと, およびポリゴンの衝突判定を行っていないことが原因で, 樹木同士や葉と枝が衝突しているように見えたり, ボクセルパーティクルと葉ポリゴンの頂点, およびエンドエフェクタの距離が近いことによる大きな力の作用 (式 2 参照) によって, 葉が変形してしまったり, 葉と枝が離れてしまったりするという不自然な現象が生じてしまう場合がある.

上記の問題点は, 本論文で提案する「大規模な森林景観」を俯瞰するようなシーンではそれほど問題にはならない (目立たない) が, カメラワークによってクローズアップになった場合などにはやはり問題となるため, カメラ位置と樹木間の距離によってシミュレーション手法を切り替える LOD (Level of Detail) を採用するなどの方策を採る必要があると考えている.

なお, 本論文で提案した手法の概要とシミュレーション結果の映像は, 参考文献リスト[20]に記載した URL にて公開しているので, 参照されたい.

6. まとめと今後の課題

本論文では, 大規模な森林景観における風による樹木の揺れをユーザが制御できるように生成する手法を提案した. 本手法では, 樹木モデル 1 本分を包含するようにバウンディングボックスを配置し, 葉 1 枚分に相当するサイズでボクセルに分割する. そして, ボクセルを構成する 4 頂点にボクセルパーティク

ルを配置し, ボクセルパーティクルを $1/f$ ノイズによって運動させることによってバウンディングボックスを変形させ, 樹木のランダムな小揺れを再現した. さらに, シミュレーション空間に樹木データを複数本配置し, ユーザがスプラインを描画することによりスプラインに沿った風を生成する手法を提案した. この風とボクセルパーティクルとの衝突判定を行うことによって, 風の影響による樹木の大揺れを再現した. ランダムな小揺れとユーザが制御可能な大揺れの 2 つを組み合わせることによって, ユーザが思い描いた「風の通り道」が見えるような印象的なシーンを生成することができることを, シミュレーション結果によって示した.

今後の課題としては, 5.2 節で述べたように, 樹木同士, およびポリゴン同士の交差・衝突を避けるための運動モデルの開発と, LOD による運動モデルの切り替えが必要であると考えている. また, 樹木の倒壊 (樹木が倒れる, 枝が折れる, および落葉) の表現への応用や, 髪の毛が風にたぶくようなシーンへの応用, およびクロスアニメーションへの応用なども挙げられる.

参考文献

- [1] Aono M, Kunii TL (1984) Botanical tree image generation. IEEE CG&A 4 (5) :10-34.
- [2] Mech R, Prusinkiewicz P (1996) Visual models of plants interacting with their environment. Proceedings of SIGGRAPH '96, pp 397-410.
- [3] Reffye P, Edelin C, Francon J, Jaeger M, Puech C (1988) Plant models faithful to botanical structure and development. Proceedings of SIGGRAPH '88, pp 151-158.
- [4] Giacomo TD, Capo S, Faure F (2001) An interactive forest. Eurographics Workshop on Computer Animation and Simulation, pp 65-74.
- [5] Sakaguchi T, Ohya J (1999) Modeling and animation of botanical trees for interactive virtual environments. Symposium on Virtual Reality Software and Technology (VRST99), pp 139-146.
- [6] Shinya M, Fournier A (1992) Stochastic motion - motion under the influence of wind. EUROGRAPHICS '92, pp C-119-C-128.
- [7] Stam J (1997) Stochastic dynamics: simulating the effects of turbulence on flexible structures. EUROGRAPHICS '97, pp 159-164.
- [8] Julien Diener, Mathieu Rodriguez, Lionel Baboud and Lionel Revert, "Wind Projection Basis for Real-Time Animation of Trees", Computer Graphics Forum (Proceedings of Eurographics 2009), Vol.28, Num.2, Mar., 2009.
- [9] Ralf Habel, Alexander Kusternig and Michael Wimmer, "Physically Guided Animation of Trees", Computer Graphics Forum (Proceedings of Eurographics 2009), 28(2) : 523-532, Mar. 2009.
- [10] S.Ota1, M.Tamura1, T.Fujimoto, K.Muraoka, N.Chiba (2004) A hybrid method for real-time animation of trees swaying in wind fields. The Visual Computer (2004) 20:613-623.
- [11] JR.Bridson, R.Fedkiw, J.Anderson (2005) Robust treatment of collisions, contact and friction for cloth animation. ACM SIGGRAPH 2005 Courses, Article No.2.

- [12]菊池司, 村岡一信, 千葉則茂 (1998) 積乱雲のビジュアルシミュレーション. 画像電子学会誌, 第27巻, 第4号, pp.317-326.
- [13]J.P.Lewis, M.Corder, N.Fong (2000) Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH 2000), pp.165-172.
- [14] Peitgen HO, Saupe D (1988) The science of fractal image. Springer, Berlin Heidelberg New York.
- [15] Wagenmakers E-J, Farrell S, Ratcliff R. (2004) Estimation and interpretation of $1/f^\alpha$ noise in human cognition. Psychonom. Bull. Rev., 11:579-615.
- [16] 武者利光 (1980) ゆらぎの世界—自然界の $1/f$ ゆらぎの不思議. 講談社, ISBN-10: 4061180428.
- [17] STAM, J. (1999) Stable Fluids. SIGGRAPH 99 Conf. Proc., Annual Conf. Series, pp.121-128.
- [18]Robert Bridson, Jin Hourihan, Marcus Nordenstam (2007) Curl-noise for procedural fluid flow. SIGGRAPH 2007 (in Proc.DVD) .
- [19]朱 暁宇, 伊藤弘樹, 菊池司, “ユーザ制御可能な風による樹木の揺れのビジュアルシミュレーション”, 第26回 Nicograph 論文コンテスト (2010年 Nicograph 秋季大会) 概要集, CD-ROM, 2010.
- [20]本論文で提案した手法の概要とシミュレーション結果の映像を公開しているURL
“ <http://www.id.takushoku-u.ac.jp/~tkikuchi/SwayingTrees/index.html> ”



朱 暁宇

2006 年中国江蘇大学計算機と技術学科卒業. 2009 年拓殖大学大学院工学研究科工業デザイン学専攻入学. 2011 年同専攻修了. 現在, アリババマーケティング株式会社勤務. 在学中は, コンピュータ・グラフィクス (CG) によるビジュアルシミュレーション, およびCG映像制作に関する研究に従事.



Abdul Hadi Bin Mohdnazir

2011 年拓殖大学工学部情報工学科卒業. 現在, 同大学院工学研究科工業デザイン学専攻博士前期課程在学中. コンピュータ・グラフィクス (CG) によるビジュアルシミュレーション, およびCG映像制作に関する研究に従事.



伊藤弘樹

1987 年専門学校桑沢デザイン研究所卒業. 同年株式会社カール事務器入社. 1992 年同社退社. 1997 年株式会社ホロンクリエイト入社. 2002 年同社退社後フリーデザイナーへ移行. 2005 年拓殖大学大学院工学研究科工業デザイン学専攻博士前期課程入学. 2007 年博士前期課程修了. 同年拓殖大学工学部デザイン学科勤務(助手). GUI デザイン及び3DCGを使用したビジュアルシミュレーションを研究. 芸術科学会, 情報処理学会, 日本デザイン学会, 他会員.



菊池 司

1999 年岩手大学大学院工学研究科電子情報工学専攻博士後期課程修了. 博士 (工学). 2000 年拓殖大学工学部工業デザイン学科 (現デザイン学科) 助手. 2004 年同大学専任講師 (現助教). 2007 年から2008 年まで韓国・高麗大学客員教授. 2009 年拓殖大学工学部工業デザイン学科 (現デザイン学科) 准教授, 現在に至る. ACM, 芸術科学会, 情報処理学会, 画像電子学会, 他会員.

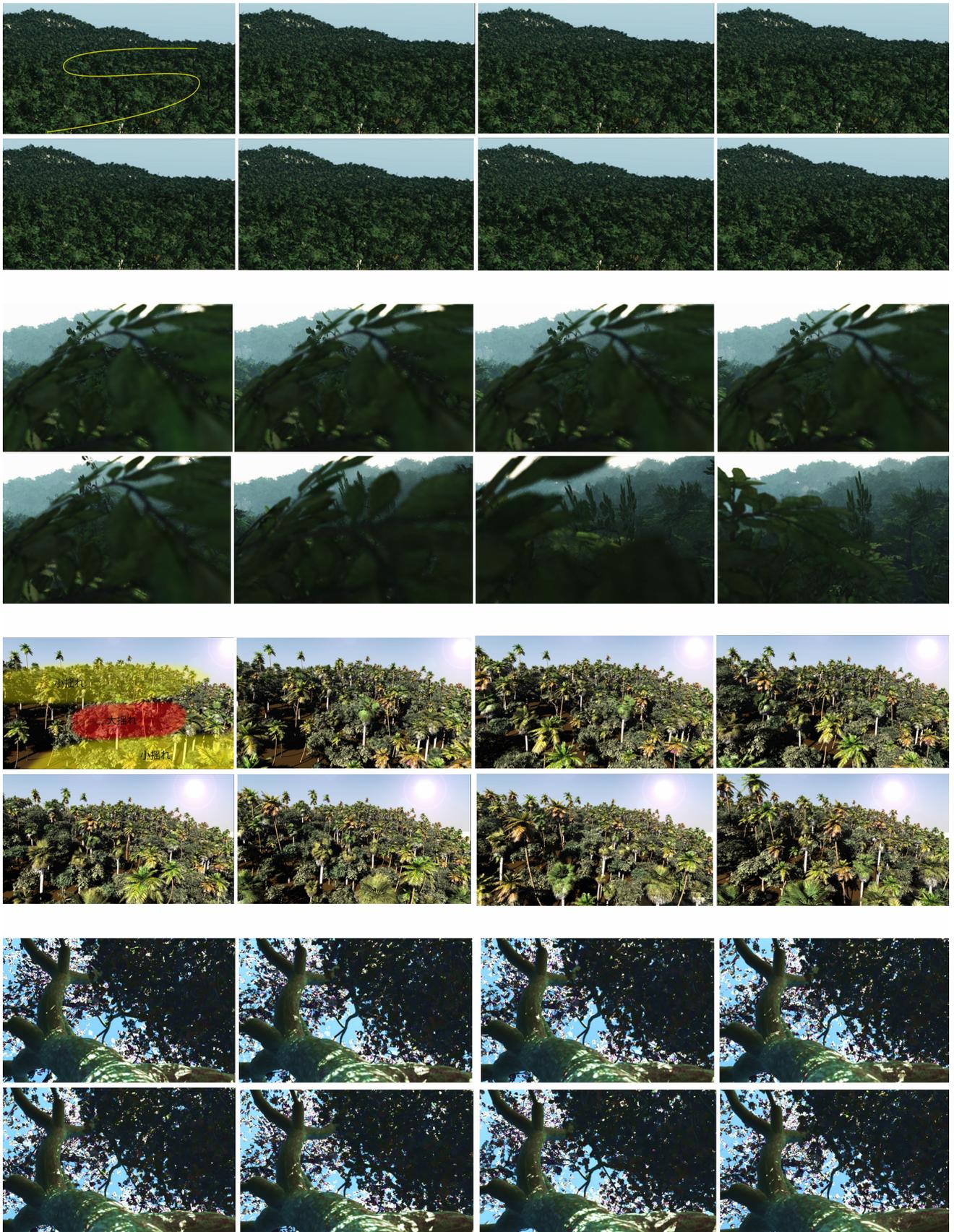


図15. 本論文で提案した手法によるシミュレーション結果の例. 図の1行目の黄色いラインはユーザが描いたスプラインの例であり, 5行目の“黄色い領域”と“赤い領域”はそれぞれ, “小揺れのみの領域”と“大揺れと小揺れを合わせた領域”であることを示している.