

## 風による樹木の揺らぎの効果音のリアルタイム生成法

松山克胤<sup>†</sup> 藤本忠博<sup>†</sup> 村岡一信<sup>‡</sup> 千葉則茂<sup>†</sup>  
<sup>†</sup>岩手大学 <sup>‡</sup>東北工業大学

### アブストラクト

筆者らは、これまで風による樹木の揺らぎの効果音を自動的に付加するサウンドモデリング法を提案している。この提案手法は、樹木は枝と葉とに分け、揺らぎの付加と効果音の生成をそれぞれ独立に行い、個々に作成された効果音を合成することで、樹木全体の効果音としている。本論文では、それぞれの効果音の生成に対して、既提案手法とは異なるアプローチに基づくリアルタイムに動作する手法を提案する。

## Real-Time Generation of Tree Movement Sound Effects

Katsutsugu MATSUYAMA<sup>†</sup> Tadahiro FUJIMOTO<sup>†</sup> Kazunobu MURAOKA<sup>‡</sup> Norishige CHIBA<sup>†</sup>  
<sup>†</sup>Iwate University <sup>‡</sup>Tohoku Institute of Technology

### Abstract

We have proposed the sound modeling method that automatically adds sound effects of tree moving in the wind. Each tree is divided into branches and leaves, and independent sound effect generation process is employed for each element. The individual results are then compounded into one sound effect. In this paper, we present a real-time generation algorithm for each process developed by employing the strategy different from previous one.

### 1. はじめに

筆者らは、樹木の枝葉の風による揺らぎのリアルタイムアニメーションを生成するためのノイズベースの効果的な手法を開発しており [Ota03], このアニメーション技術に対して、さらに風による揺らぎの効果音を自動的に付加するサウンドモデリング法を提案した [松山 02, 松山 03]. その提案手法では枝の効果音はカルマン渦列の周波数に基づいた手法を、葉の効果音は葉同士の位置関係から波形表を作成する手法を提案している。本論文では、既提案手法とは異なるアプローチによる、リアルタイムに動作するサウンドモデリング法を提案する。本提案手法は、既提案手法とはアプローチが異なるためそれぞれに特徴がある。既提案手法では、計算時間をさらに必要とするが、より正確な交差判定を行うことにより、モーションと一致した接触音の生成が可能となるため近接シーンの表現も可能となる。また、前計算を必要としないため使用メモリを増加させることなく、多種多様な樹木からなるシーンのウォークスルーなども可能となる。一方、今回の提案手法では、交差判定を統計的に行うことにより、インタラクティブな応用を可能としているため、特に遠景のシーンの効果音の生成に向く。

一般に、音楽CDレベルの高品質な音はサンプリングレートが 44100Hz 以上で成立しうるが、本論文で

はリアルタイムにこれを満足する効果音を生成する手法を提案する。また、フレームレートは可変で、リソースに応じて自動的に対応させている。

### 2. 関連研究

コンピュータグラフィックスの分野における効果音の自動生成に関する研究の先駆的な試みは 10 年前にさかのぼるが [Takala92], その後の研究例はほとんどなく、ここ 2, 3 年で活発になってきた新しい研究分野である [Dobashi03, Doel01, Dubnov02, O'Brien01, 土橋 03]. 3次元仮想環境での音生成の技術は、物体の形状や特性から、その物体が音源として発生し得る波形の構築をサウンドモデリング、音源から受聴点に届く反響の計算によって求められる音響効果としてのフィルタの生成であるサウンドレンダリングとに大別できる [Takala92].

サウンドモデリングに関する研究は、物理法則に基づいた、衝突、摩擦、回転に対して効果音を自動的に付加する手法 [Doel01] が提案されている。これはインタラクティブかつ自動的に効果音を付加できる先駆的な研究である。この研究に関連して、さらに、現実の 3D オブジェクトの音を実測すること [Pai01] により、自然な音を作成することも可能となっている。また、非線形有限要素法 [O'Brien99] を用いた、短いタイムス

トップで幾何形状の微小変化を利用して音を作成する手法[O'Brien01]により、物理的に説得力がある、自然な効果音の自動生成が可能となった。これは計算コストが大きいという問題点はあるものの、非常に興味深い研究である。そして、ウェーブレット解析を静的に行い、キャラクタやテキストに投影することで新しい音を再構築する手法[Dubnov02]が提案されている。

枝の風きり音に関連した研究としては、土橋等により、サウンドテキストチャーを用いた風きり音のリアルタイムレンダリング法が最近提案されている [Dobashi03, 土橋 03]。筆者等は、樹木のように多数の構成要素からなる物体を想定して手法開発を行っているため、両者のアプローチは異なっている。

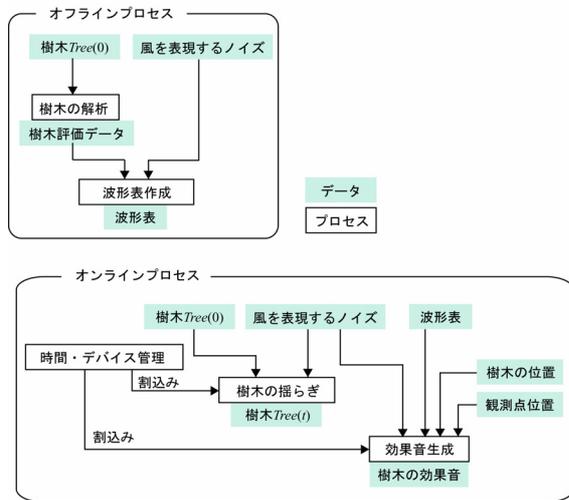


図 1 : システム構成図

### 3. システム構成

本論文は、樹木  $TREE$  は、枝と葉とで構成されていると考える。

$$TREE(t) = \{B(t), L(t)\}$$

$B$ : 枝の集合

$L$ : 葉の集合

時刻  $t$  は  $t \geq 0$  であり、 $t=0$  で無風状態を示すものとする。また、枝葉ともに要素に番号を割り振る。番号は一意で要素数を超えない値である。時刻  $t$  の  $i$  番目の枝は  $B_i(t)$ 、葉は  $L_i(t)$  と記述する。

本提案手法はオフラインプロセスとオンラインプロセスとに分かれている (図 1)。オフラインプロセスは樹木の形状データを入力とし、解析を行うブロックと、波形表を作成するブロックとで構成される。オフラインプロセスで作成された波形表は、オンラインプロセ

スの効果音生成の入力に使用され、時間・デバイス管理ブロックからの割り込み信号で作動する効果音生成ブロックで再生される。同じブロックでも、枝と葉は別々の手法で解析や波形表作成を行う。枝の効果音生成を第 4 章に、葉の効果音生成を第 5 章に、実行結果を第 6 章に、第 7 章にまとめと今後の課題を、それぞれ記述する。

### 4. 枝の効果音生成

筆者等はこれまで、枝の効果音の生成法として、実験的に求められているカルマン渦列の周波数を基底周波数とし、それに音色を付加する手法を提案している [松山 02]。この手法は以下のようなものである。カルマン渦列の周波数  $f_0$  は、円柱または円錐台の直径  $D$  と流速  $U$  とを用い、以下の式で記述される。

$$f_0 = StU / D \quad (1)$$

$St$ : ストローハル数

ストローハル数  $St$  は、0.2 という定数として扱う。また、 $f_0$  の音の強さ  $I_0$  は、

$$I_0 \propto U^6 / r^2 \quad (2)$$

$r$ : 枝と観測点との距離

となる。これを枝の両端それぞれに適用し、補間を行うことで枝全体の基底の音とする。これにより求められた  $f_0$  及び  $I_0$  から、以下の式で音色を付加する。

$$I = I_0 / (f / f_0)^\beta \quad \left( f_0 < f < \frac{f_s}{2} \right) \quad (3)$$

$f_s$ : サンプリング周波数

$\beta$ : 音色を決定する定数

また、窓関数にガウス関数を使用することで、パルス状のノイズを防いでいる。ガウス関数は次の式で表される。

$$Gaussian(x) = ce^{iax} e^{-\frac{(x-b)^2}{4a}} \quad (4)$$

ガウス関数は複素関数であるので、以下の係数を設定し、実数関数にする。

$$c=1, a=0, b=0, \alpha=1 \quad (5)$$

そして、さらに、時刻  $t$  の前後数ステップの波形も用い、これらに窓関数を掛け、足し合わせる。これはパルス状のノイズを防ぐためである。また、このプロセスは、樹木から発生する音そのものを計算しているため、式(2)の枝と観測点との距離  $r$  はサウンドレンダリング部によって減衰させる。すなわち、式(2)の  $I_0$  は  $U$

の関数である。このプロセスを一般的な PC を用い、フレームステップ(30fps)でリアルタイムに行うことは現状では困難である。

式(1)は流速  $U$  と枝の直径  $D$  に、式(2)は流速  $U$  に依存する数式であるが、式(1)の  $D$  を前もって決定することで、式(1),(2)ともに  $U$  のみに依存する数式とする。流速  $U$  と直径  $D$  を離散的にすることで、流速  $U$  に関する波形表を前処理で作成し、リアルタイムに再生することが可能となる。ここで、複数の枝をひとまとめにする単位は空間の音源の位置情報をまとめることであり、本論文では樹木一本を一つの音源と考え、これを評価する。

式(1)で、枝の根元と先端それぞれの位置で発生する周波数を求めているが、この端点での周波数の差を無視し、根元と先端で同じ周波数とすることを考える。この近似を行うことで、枝の評価に枝の長さを直接利用できる (図2)。

オンラインのプロセスは、入力された風速  $U$  を離散化し、その波形表を再生するだけとなる。波形表の再生の際、現在の再生ポイントを示すレコードの「針」を用いることで、連続性を確保する。これは再生時のノイズの混入を防ぐためであるが、 $U$  の変化に伴う波形表の変化で混入されるノイズは除去できない。これは前ステップ時の波形表及びその「針」を用い、クロスフェードさせることで解決する。

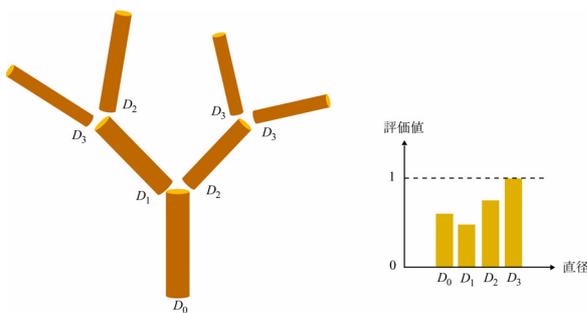


図2：枝の評価

## 5. 葉の効果音生成

葉の効果音は葉身同士の衝突・摩擦をシミュレーションする手法[松山 03]で、音源として複数の波形表を作成し、個々の葉に対して葉身の大きさに依存して波形表のひとつを割り当て、それを  $L_i$ ,  $L_j$  の相対的な位置関係の状態を基に再生するものである。

本論文では、葉群の動きを統計的に示す、「乱雑さ

(entropy)」のパラメータを用いることで葉の効果音を生成する手法を提案する。乱雑さは、対象の葉群の揺れの度合いを示すもので、乱雑さが大きいほど揺れが激しいことを示し、0～1の値をとる。効果音生成には、この値は主に葉同士の単位時間内の衝突数、および衝突時の強さに影響する。乱雑さを離散化することで前処理が可能となり、高速化を行うことができる。また、本論文では簡単のため、1本の樹木を構成する葉群を、1つの乱雑さを用いる単位とする。

前処理は大きく2つのブロックに分かれる。最初に入力された樹木の葉を解析し、基となる波形表を作成するブロックで、2つめは、基となる波形表を用い、乱雑さに応じて再生する波形表を作成するブロックである。初めのブロックでは、まず、入力された葉を解析し、葉柄の最大・最小値を決定し、これを基に葉柄の長さの区分けを行う。葉の大きさは葉柄の長さに比例させているため、これは葉の持つ基底周波数での区分けとなる。これで対象の葉群の統計的特徴を葉の大きさとまとめ、これを評価することで基底周波数群を決定する。この基底周波数から、周波数スペクトルがピンクノイズとなるような音色を作成し、逆フーリエ変換することで時間領域の波形を得る。ピンクノイズは枝の音色の決定と同様に、

$$I = (f / f\_basis)^\beta$$

$I$ : 音の強さ  
 $f$ : 周波数  
 $f\_basis$ : 基底周波数  
 $\beta$ : 音色パラメータ

で決定する。これで基となる波形表を得る。この波形表は、葉群が固有に持ち得る波形である。そして、次のブロックで、乱雑さに応じて再生される波形表を作成する。これは乱雑さの刻み数だけ生成される。乱雑さと作成する波形表との関係は表のようになる。単位時間内の衝突数 ( $numKeyOn$ ) は乱雑さと対象葉群の密度 ( $density$ ) により決定する。

$$numKeyOn \propto entropy \cdot density$$

衝突時の音の強さ ( $envelope$ ) は乱雑さに依存させる。

$$envelope \propto entropy$$

また、衝突が摩擦に移行する場合の減衰や、摩擦状態の減衰も乱雑さに依存し、

$$envelope = envelope \cdot entropy$$

とする。衝突・摩擦状態が終了する数 ( $numKeyOff$ )

は衝突・摩擦状態の継続条件に関連するものとする。  
継続条件は、2つの乱数  $rand_1, rand_2$  を用い、

$$entropy \cdot rand_1 \leq rand_2 \quad (10)$$

が真で継続、偽で終了する。これは乱雑であるほど継続せず、 $numKeyOff$ が増加する。

乱雑さ  $entropy$  をどのように決定するかは、葉の揺らぎ部分と密接に関係している。本論文では、葉の揺らぎにローカルなノイズとグローバルなノイズと2つ使用している。時刻  $t$  に、ある程度整列された葉  $L_i(t)$  の軸  $S$  に対して適用される揺れ幅  $\theta_i(t)_S$  は、ローカルなノイズとグローバルなノイズをそれぞれ  $Local_i(t)_S, Global_i(t)_S$  として、

$$\begin{aligned} \theta_i(t)_x &= Local_x(t-i) \cdot Global_x(t) \\ \theta_i(t)_y &= Local_y(t-i) \cdot Global_y(t) \\ \theta_i(t)_r &= Local_r(t-i) \cdot Global_r(t) + \alpha \theta_i(t)_x \end{aligned} \quad (11)$$

となる。ローカルなノイズは個々の葉に各々の値が振られるが、樹木全体で、ローカルなノイズの1周期を超える場合もある。この場合は従ってローカルなノイズは乱雑さを示すものではなく、樹木全体の葉が同一に揺らぐのを防ぐ役割の比重が高い。逆にグローバルなノイズは樹木全体のダイナミクスを表し、これが乱雑さを表現していると考えられるため、乱雑さ ( $entropy$ ) はグローバルなノイズ値のみを用いて決定する。

オンラインのプロセスは、入力された乱雑さを離散化し、その波形表を再生するだけとなる。枝の音生成と同様に、針を用いることで連続性を確保し、クロスフェードを行うことでノイズの混入を防ぐ。

## 6. 実行例

上記のアルゴリズムを、2.53GHz Pentium(R)4 CPU 1.5Gbyte RDRAM メモリ上でC++を用いて実装した。図3は実行時のスクリーンショットを示す。本論文では、1本の樹木を1つの音源とみなす。表1は、計算量の比較のために用いた2種類の樹木データおよび、揺らぎのみを実行したときのフレームレートと平均CPU利用率を示す。CPU利用率は時間管理プロセスのタイマーによって、ある程度CPUをブロックさせているので、このブロックを除去した場合の平均フレームレートは、表1の平均フレームレート×100÷表1の平均CPU利用率となる。

枝の効果音生成は、直径  $D$  の分割数と流速  $U$  の分割

数が、また、葉の効果音生成は、葉柄の長さ  $L$  の分割数と乱雑さ  $E$  の分割数が、音質に関するパラメータとなる。これらのパラメータは大きいほど良い結果が期待されるが、あまり大きすぎると、処理に時間がかかり、使用するメモリサイズが増大する。分割数は大きすぎてもその効果が知覚できないので、計算環境に依存した分割数を決定する必要がある。本章ではこれらの分割数を、8, 16, 32, 64, 128, 256に設定し、各々の組み合わせに対する計測を行う。なお、筆者らが感覚的に自然であると感じた最小限のパラメータは、直径  $D$  の分割数が16、流速  $U$  の分割数が128、葉柄の長さ  $L$  の分割数8、乱雑さ  $E$  の分割数128である。以下のサブセクションでは、上記のパラメータの分割数に関して、効果音生成に必要とした時間を計測した。

	樹木A	樹木B
枝の本数	123	1160
葉の枚数	4860	16380
平均フレームレート[fps]	21.3	6.4
平均CPU利用率[%]	71	91

表1：樹木データと揺らぎのフレームレート



図3：スクリーンショット  
上：樹木A、下：樹木B

## 6. 1 枝のオフラインプロセスの時間計算量

枝の効果音生成のオフラインプロセスは、枝データの解析と波形表生成とに分かれている。枝データ解析プロセスの計算時間は樹木 A,B 共に 1ms 弱であったので計測の対象外とする。表 2 は波形表作成に関する時間計算量である。樹木の大きさに関わらず、直径  $D$  の分割数と流速  $U$  の分割数に比例しているといえる。

枝: 123		流速の分割数						
		8	16	32	64	128	256	
直径の分割数	8	6.266	12.484	24.922	54.875	101.672	202.781	
	16	12.437	24.531	49.422	106.875	200.109	399.219	
	32	24.422	48.922	98.078	207.906	395.031	788.312	
	64	48.953	97.000	194.203	403.469	780.344	1558.890	
	128	98.781	192.640	385.344	787.891	1545.219	3098.515	
	256	192.547	383.812	767.391	1552.672	3074.719	6145.609	

表 2 : 分割数と計算時間 (S)

葉: 4860		乱雑さの分割数						
		8	16	32	64	128	256	
葉柄の分割数	8	0.781	0.782	0.797	0.860	0.797	0.797	
	16	1.562	1.610	1.609	1.718	1.609	1.609	
	32	3.219	3.203	3.203	3.391	3.234	3.218	
	64	6.454	6.406	6.406	6.391	6.422	6.406	
	128	12.891	12.812	12.812	13.000	12.828	12.812	
	256	25.657	25.578	25.578	25.796	25.625	25.609	

葉: 16380		乱雑さの分割数						
		8	16	32	64	128	256	
葉柄の分割数	8	0.875	0.781	0.812	0.812	0.828	0.813	
	16	1.719	1.641	1.625	1.609	1.703	1.625	
	32	3.454	3.266	3.218	3.234	3.313	3.219	
	64	6.812	6.500	6.438	6.438	6.578	6.569	
	128	13.266	12.875	12.812	12.813	12.969	13.043	
	256	16.109	25.594	25.547	25.531	25.672	25.821	

表 3 : 葉の大きさに依存した波形表作成

葉: 4860		乱雑さの分割数						
		8	16	32	64	128	256	
葉柄の分割数	8	8.094	17.000	34.985	71.000	142.843	286.703	
	16	8.031	17.000	35.015	70.985	142.828	286.875	
	32	8.031	17.015	34.937	71.031	143.078	286.734	
	64	8.046	17.000	34.922	70.968	143.875	286.922	
	128	8.000	17.000	35.016	71.000	142.828	286.797	
	256	8.047	17.016	34.953	71.032	142.859	286.844	

葉: 16380		乱雑さの分割数						
		8	16	32	64	128	256	
葉柄の分割数	8	73.594	154.719	318.344	646.469	1298.297	2607.953	
	16	73.156	154.922	318.453	646.125	1299.375	2611.938	
	32	73.093	154.968	318.578	646.110	1299.203	2610.656	
	64	73.125	154.875	318.360	646.125	1302.609	2625.020	
	128	73.110	154.844	318.625	646.515	1300.094	2628.746	
	256	73.078	154.907	318.594	646.375	1300.766	2637.343	

表 4 : 実際に再生される波形表作成

## 6. 2 葉のオフラインプロセスの時間計算量

葉のオフラインプロセスは、葉データの解析、葉の大きさに依存した波形表の作成 (LP1)、そして、キー・オン数、継続条件等を用いた、実際に再生する波形表の作成 (LP2) の 3 つに分けられる。枝と同様に、解析プロセスの計算時間は樹木 A,B 共に 1ms 弱であったので計測の対象外とする。表 3 は LP1 の、表 4 は LP2 に要した時間計算量である。

LP1 は葉柄の分割数に、LP2 は乱雑さの分割数にそ

れぞれ依存しており、LP1 より LP2 の方が支配的である。LP1 の葉の枚数による計算時間の増大は、葉の大きさの違いから来ていると考えられる。樹木 A より樹木 B の葉の方が大きいので、作成される波形表の基底の周波数は低くなる。したがって、樹木 A より樹木 B の方が、より多くの代入操作の必要があり、それが時間計算量に影響していることが考えられるが、それほど大きい影響ではない。LP2 に関する時間計算量は、葉の密度の違いに関連している。乱雑さが同じなら、密度が高いほどキー・オン数が多くなり、結果的に多くの計算を必要とする。

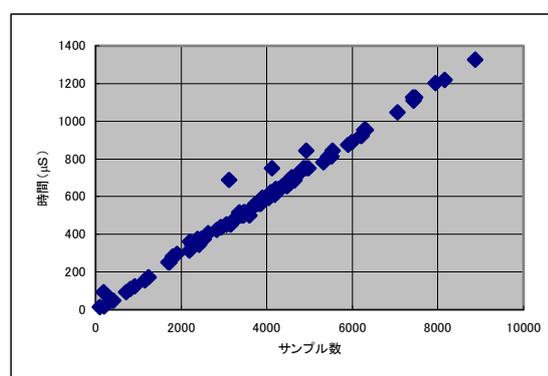


図 4 : 効果音生成時間計測

## 6. 3 効果音生成の時間計算量

効果音生成プロセスは、時間・デバイス管理プロセスの割込み時に作成すべきサンプル数を受け取り、その数だけ波形を作成する。入力 of 樹木が複数である場合、また、一つの樹木に複数の音源を配置した場合、リアルタイム性を維持するために、このプロセスがどの程度の音源に対してリアルタイムに処理できるかが論点となる。図 4 は、作成すべきサンプル数に対し、処理に要した時間をプロットしたものである。この図はほぼ直線であり、1 秒 (44100Hz) の音生成に対し、平均で約 7ms 要している。したがって、上記の計算機に効果音生成のみを処理させた場合、140 程度の音源を配置できる。

## 6. 4 領域計算量

本論文では、枝葉ともに波形表を用いることでリアルタイムな音生成を可能としている。波形表に必要なメモリのサイズは、枝の波形表は流速  $U$ 、葉の波形表は乱雑さ  $entropy$  それぞれの分割数  $n$  に依存する。本

論文では、波形表のサンプル数を 44100、サンプルを倍精度浮動小数点型 (64bit) で表現しており、枝葉共に使用するメモリのサイズは、

$$44100[\text{sample}] \cdot 64[\text{bit}] \cdot n[\text{分割数}] / 8[\text{bit/byte}] \quad (12) \\ = 0.3528n[\text{Mbyte}]$$

となる。実装する場合には利用可能なメモリ領域内に収まるように分割数を設定する必要がある。

## 7. まとめと今後の課題

本論文では、樹木に対し、風による揺らぎによる効果音を自動的かつリアルタイムに付加するサウンドモデリング法を提案した。これにより、VR環境などにおいて、インタラクティブに動作する樹木の揺らぎに、樹木の効果音を付加することも可能となった。

本提案手法は、枝と風による音と葉同士のぶつかりあう音をシミュレーションするものであるが、実際にはさらに、枝同士の衝突、枝と葉との衝突、葉と風による音、そして、葉が飛ぶときに鳴る音がある。これらの要因を考慮することで、よりリアルな樹木の効果音が作成できると考えられる。また、本手法では広葉樹を想定しており、針葉樹への拡張も興味深い。さらに、風の強さについては日常的なレベルを想定しており、嵐のような風への拡張も興味深い。

## 参考文献

- [Dobashi03] Dobashi, Y., Yamamoto, T., Nishita, T., Real-Time rendering of aerodynamic sound using sound textures based on computational fluid dynamics, *SIGGRAPH2003*, ACM, 2003
- [Doel01] Doel, K., Kry, P. and Pai, D. Foley automatic: Physically-based sound effects for interactive simulation and animation, In *Proceedings of ACM SIGGRAPH*, ACM, 2001; 537-544.
- [Dubnov02] Dubnov, S., Joseph, Z., Yaniv, R., Lischinski, D. and Werman, M. Synthesizing sound textures through wavelet tree learning, *IEEE Computer Graphics and Applications* 22, 4, 2002; 38-48.
- [O'Brien99] O'Brien, J. and Hodgins, J. Graphical modeling and animation of brittle fracture, In *Proceedings of ACM SIGGRAPH99*, ACM, 1999; 137-146.
- [O'Brien01] O'Brien, J. Cook, P. and ESSL, G. Synthesizing sounds from physically based motion, In *Proceedings of*

*ACM SIGGRAPH2001*, ACM, 2001; 529-536.

[Ota03] Ota, Tamura, Fujita, Muraoka, Fujimoto, and Chiba, "1/f<sup>β</sup> Noise-Based Real-Time Animation of Trees Swaying in Wind Fields," *Proceedings of the Computer Graphics International 2003 international conference*, 2003.

[Pai01] Pai, D., Doel, K., James, D., Lang, J., Lloyd, J., Richmond, J. and Yau, S. Scanning physical interaction behavior of 3D objects, In *Proceedings of ACM SIGGRAPH2001*, ACM, 2001; 87-96.

[Takala92] Takala, T. and Hahn, J. Sound Rendering, In *Proceedings of ACM SIGGRAPH92*, ACM, 1992; 26: 2: 211-220.

[土橋 03]土橋, 山本, 西田, 流体力学に基づくサウンドテクスチャーを用いた風きり音のリアルタイムレンダリング, *Visual Computing / グラフィクスと CAD 合同シンポジウム 2003*

[松山 02] 松山, 藤本, 村岡, 千葉, 風による樹木の揺らぎの効果音の生成法, 第 18 回 NICOGRAPH 論文コンテスト, 芸術科学会, pp. 41-46, 2002.

[松山 03] 松山, 藤本, 村岡, 千葉, 風による樹木の揺らぎの効果音の生成—葉の揺らぎの効果音—, 2003 年 NICOGRAPH 春季大会, 芸術科学会, pp. 67-68, 2003.