

# ろうそくの炎の静止画像から炎の揺らぎアニメーションを 生成するノイズベースモーフィング

小川 優樹<sup>†</sup> 松山 克胤<sup>‡</sup> 藤本 忠博<sup>‡</sup> 千葉 則茂<sup>‡</sup>

<sup>†</sup>株式会社プレミアムエージェンシー      <sup>‡</sup>岩手大学

## Noise-Based Morphing for Producing Flickering Candle Flame Animation from a Single Still Image of Flame

Yuuki OGAWA<sup>†</sup> Katsutsugu MATSUYAMA<sup>‡</sup> Tadahiro FUJIMOTO<sup>‡</sup> Norishige CHIBA<sup>‡</sup>

<sup>†</sup>Premium Agency Inc.    <sup>‡</sup>Iwate University

kmatsu @ iwate-u.ac.jp

### アブストラクト

本論文では、炎のスタイル（2次元の静止画像）に対して、Beierらのモーフィング法を適用することで、炎の揺らぎアニメーションを生成する手法を提案する。このモーフィング法では、原画像と目的画像領域に、対応する制御線を配置することにより、画像の変形を指定する。ここでは、風や空気の乱れを $1/f^\beta$ ノイズで表すことができると仮定し、制御線に揺らぎ運動を与えている。本手法により、軽風で観察される炎の揺らめきの表現が可能となり、また、複数のろうそくの炎の揺らぎの表現をリアルタイムに行うことが可能となった。

### Abstract

In this paper, we propose the technique for animating flickering flames by applying Beier et al's morphing method to a single still image of a candle flame. In this morphing method, a pair of control lines arranged on a source image and a destination image area are used to define the deformation of the source image. Assuming that the fluctuation of wind can be expressed by  $1/f^\beta$  noise, we apply the noise to the control lines for giving natural fluctuation. Our method can produce an animation of not only a candle flame flickering in light breeze but a scene including many candles.

## 1 はじめに

炎や水などの流体に関するビジュアルシミュレーションは、長い間研究者の興味を集めてきた研究テーマの1つである。これまでに提案されている炎のアニメーション生成手法は、高い写実性を実現するために、物理シミュレーションベースのアプローチをとっているものが多い[1-3]。一方、バーチャルリアリティやコンピュータゲームなどのインタラクティブシステムにおいては、大規模な現象をリアルタイムに表現することができるアニメーション技術が期待されている。このような目的には、物理シミュレーションベースの手法に比べて、遥かに効率的な手法が必要となる。

近年、実用性を重視した映像生成手法として、ノイズを用いた揺らぎアニメーション生成[4-7]や、ビデオ映像のループ化[8]、ビデオデータベースを用いた流体のアニメーション生成[9]などの、現象面からのアプローチが行われている。

本研究は、ろうそくの炎を対象として、炎の揺らぎアニメーションを生成するものである。ろうそくを大量に配置するなどの大規模な表現をリアルタイムで行うことを目的とし、そのために現象面から炎のゆらぎを表現していくアプローチをとる。

本論文では、炎のスケル(2次元の静止画像)に対して、Beierらのモーフィング法を適用することで、炎の揺らぎアニメーションを生成する手法を提案する。このモーフィング法では、原画像と目的画像領域に、対応する制御線を配置することにより、画像の変形を指定する。ここでは、風や空気の乱れを $1/f^\beta$ ノイズで表すことができると仮定し、制御線に揺らぎ運動を与えている。なお、筆者らの一部は、ノイズとモーフィングを用いて、ろうそくの炎の揺らぎアニメーション生成を試みている[7]が、本論文で提案する手法はその大幅な拡張である。

本手法により、軽風で観察される炎の揺らめきの表現が可能となり、また、複数のろうそくの炎の揺らぎの表現をリアルタイムに行うことが可能となった。

### 1.1 関連研究

本節では、炎のアニメーション生成およびノイズベースのアニメーション生成に関する先行研究を挙げる。

**炎のアニメーション生成:** これまでに提案されている炎のアニメーション生成手法は、高い写実性を実現するために、物理シミュレーションベースのアプローチをとっているものが多い[1-3]。代表的なものに、粒子による爆発火炎の表現法[1, 2]、空間をボクセルに分割する計算手法[3]などが挙げられる。現象面からのアプローチ手法に、ビデオデータベースを用いて、1枚の画像の炎領域に適合するアニメーションを生成する手法がある[9]。この手法は様々な流体の表現を可能とするが、本研究の目的であるリアルタイムアニメーションを生成することはできない。

**ノイズベースのアニメーション生成:** 自然現象の揺らぎを $1/f^\beta$

ノイズで表すアニメーション手法がいくつか提案されている。代表的なものとして、風による樹木の揺らぎ[4]、草原のような線状物体の揺らぎ[5]、および、旗やカーテンのような面状物体の揺らぎ[6]への適用が挙げられる。また、筆者らの一部は、ノイズとモーフィングを用いて、ろうそくの炎の揺らぎアニメーション生成を試みており[7]、本論文で提案する手法はその大幅な拡張である。

## 2 $1/f^\beta$ ノイズの性質と生成

$1/f^\beta$ ノイズは、図1に示すように、パワースペクトルが周波数 $f$ の $\beta$ 乗に反比例するノイズをいう。 $\beta$ はノイズの特徴を表し、値が小さいほどノイズの変化が激しく(相関が小)、値が大きいほどノイズの変化が緩やか(相関が大)であるという性質がある。 $\beta$ が0であるとき、ノイズは白色ノイズと呼ばれ、すべての周波数成分を同じ大きさで含んでいる。また、 $\beta$ が1のときはピンクノイズ、 $\beta$ が2のときはブラウンノイズと呼ばれる。

$1/f^\beta$ ノイズは、揺らぎ現象として、風や炎のような流体現象や、地形、生体現象といった多くの自然現象から観測されている[10]。本手法では、風で生じる空気の乱れを $1/f^\beta$ ノイズで表すことができると仮定してろうそくの炎の揺らぎを表現する。

本手法では1次元ノイズで風を表現する。ノイズは、高速逆フーリエ変換を用いたスペクトル合成法[11]により生成する。スペクトル合成法を簡単に説明する。

周波数 $0 < k < N/2$ に対して、共役すなわち、 $F(k) = F^*(N - k)$ となるような周波数スペクトルを計算する。

$$F(k) = S_k \cos(\omega) + jS_k \sin(\omega) \quad (1)$$

$$S_k = C / k^{\beta/2}$$

$$\omega = 2\pi \cdot \text{rand}$$

ここで、 $C$ は定数、 $\text{rand}$ は区間[0, 1]の一様乱数である。この $F(k)$ を高速逆フーリエ変換することで、 $1/f^\beta$ ノイズを得る。

これにより生成されたノイズは周期性を持つため、リピート再生のように繰り返しノイズを参照することができる。

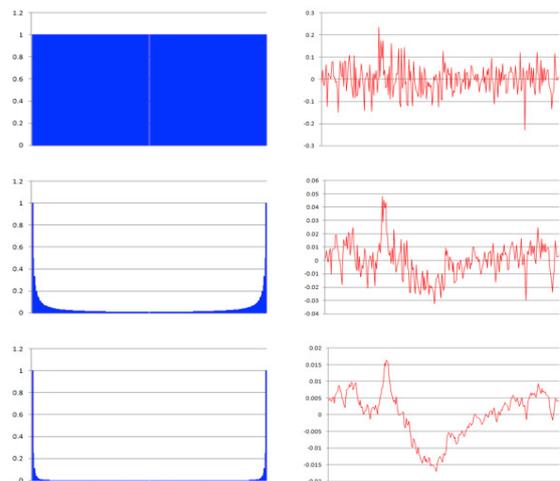


図1  $1/f^\beta$ ノイズ。  
左はパワースペクトル、右は波形。上から、 $\beta=0, 1, 2$ 。

### 3 モーフィング

本手法は、炎のスタイル（2次元の静止画像）に対してモーフィングを適用することで、揺らぎのアニメーションを生成する。本論文で用いるモーフィングは、文献[12]のモーフィング手法の高速化を行ったものである。

文献[12]の手法は、図2に示すように、原画像と目的画像に配置した制御線を用いて、変形を行う手法である。変形のアルゴリズムを以下で簡単に説明する。

目的画像の制御線 $PQ$ と原画像の制御線 $P'Q'$ を用いて、目的画像上の画素 $X$ に対応する原画像上の画素 $X'$ を以下のように計算する（図3）。

$$u = \frac{(X-P) \cdot (Q-P)}{\|Q-P\|^2} \quad (2)$$

$$v = \frac{(X-P) \cdot \text{Perpendicular}(Q-P)}{\|Q-P\|}$$

$$X' = P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|}$$

ここで、 $\text{Perpendicular}()$ は、入力ベクトルに直交する、長さが等しいベクトルを返す関数である（直交するベクトルは、左向きと右向きの2つあるが、一貫していればどちらを選択しても良い）。なお、制御線が複数ある場合、各制御線について対応する画素の計算を行い、この各画素の重み付き平均を計算することで、対応する位置を求める。重みは、画素と制御線との距離、制御線の長さなどが用いられる。

実際の変形は、原画像の各ピクセルが、目的画像のどこに移動するのかを計算する（前方マッピング）のではなく、目的画像の各ピクセルの画素値を、対応する原画像から求める（後方マッピング）。これは、変形による原画像ピクセルと目的画像ピクセルが画素レベルで1対1対応しないことが原因である。後方マッピングでは、対応する原画像の位置の近傍4ピクセルを用いて補間することで画素値を得る。この手法では、目的画像の全ピクセルの画素値の計算をおこなっているため、計算コストが高い。

そこで本手法では、計算コストを抑えるために、図4(a)に示すような原画像を、図4(b)に示すように四角形のメッシュに分割し、そのメッシュの格子点のみを、移動の計算に利用することで、目的画像に配置した制御線（緑線）が導く位置へと移動させ（図4(c)）、そうして得られた新たな座標値を持つ格子点で構成される四角形メッシュに、原画像をマッピングすることで（図4(d)）、目的画像（図4(e)）を得る方法を用いる。この、前方マッピングによる方法により、簡単なシェーダの実装でGPUを用いたテクスチャマッピングが行えるため、CPU使用率を減らすことができ、計算コストを抑えることができる。

本手法では、メッシュの分割数を適切に設定する必要がある。計算量と画像の品質との間にはトレードオフがあり、実験により8x8のメッシュ分割（格子点は64）が適当であると判断した。

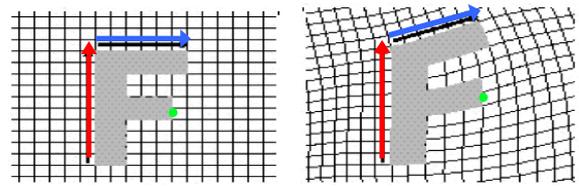


図2 モーフィング。  
配置した制御線に従って画像が変形する。

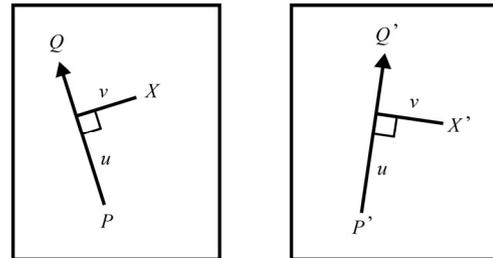
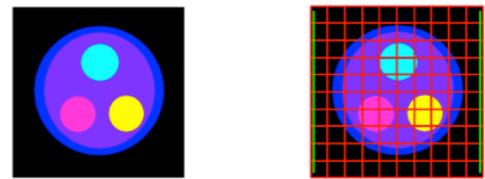
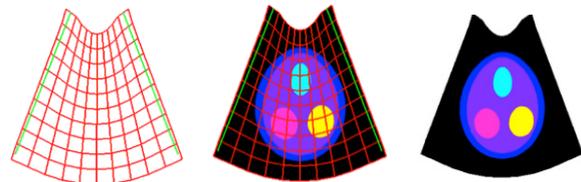


図3 モーフィングアルゴリズム。  
左が目的画像、右が原画像。

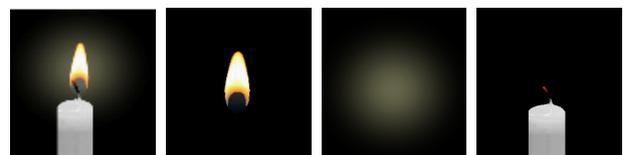


(a) 原画像 (b) メッシュ状に分割



(c) 格子点の移動 (d) マッピング (e) 目的画像

図4 本手法のモーフィング



(a) モデル (b) 炎 (c) 炎が発する光 (d) ろうそく

図5 ろうそくと炎のモデル

これ以下の分割では、目的画像に生じる歪みが顕著であり、また、これ以上の分割による結果の改善はわずかであった。例えば、512x512[pixel]の原画像を8x8のメッシュで分割した場合、対応点の計算は、 $(8 \times 8) / (512 \times 512) = 1 / 4096$ で済む。

## 4 ろうそくと炎のモデル

本手法は、カメラ撮影で得られた炎のスタイルを用いる。炎の揺らぎはモーフィングにより行われるために、図5(a)のようなスタイルをそのまま用いると制御線の影響が光やろうそくにも適用され、不自然な結果となる。そこで、炎のみに揺らぎを適用するために、炎の部分のみを切り出して使用する。ろうそくの炎の表現に、炎のスタイル (図5(b)), 炎が発する光 (図5(c)), および、ろうそくの3次元モデル (図5(d)) の合成により、ろうそくと炎を (図5(a)) 表現する。

炎のスタイルおよび炎が発する光は、ビルボードのように視点方向に向けて配置する。これを基本型とし、この状態から揺らぎを与える。

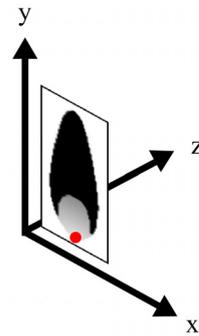


図6 ビルボード。  
赤い点が回転の中心となる。

## 5 揺らぎ現象に基づいた炎の変形

前章で述べた炎のスタイルに対してモーフィングを適用することで、揺らぎのアニメーションを生成する。炎の揺らぎをモーフィングの制御線の移動により表現するが、ここでは空気の乱れを  $1/f^\beta$  ノイズで表すことができると仮定して制御線の移動を行う。

本研究では、実際のろうそくの炎を観察し、現象面から分析を行った。その結果を受けて、炎の揺らぎを、閉め切った部屋などで見られるような平穏な状況での揺らぎ、および、軽風により生じる揺らぎに分けて、それぞれ独立して生成することとした。そして、それぞれの揺らぎを組み合わせることで、最終的な炎の揺らぎを表現する。なお、平穏および軽風は、ポーフォート風力階級で定義されている名称であり、ここでは、風の程度を表す言葉として参照的に使用する。

平穏での揺らぎは軽風での揺らぎに比べて穏やかである。平穏での揺らぎは無指向性の揺らぎであり、軽風による揺らぎは指向性の風による揺らぎであるとする。また、炎は常に平穏での揺らぎの影響下にあるが、軽風は一時的に生じる揺らぎを指し、この違いは、風の強さが周期性を持って適用されるかどうかの違いとして表現される。

なお、前章で述べたように、静止状態の炎を、視点方向に向けて配置したビルボードの状態であるとする。ここでは説明のために、視点からビルボード方向をz軸、ビルボードの鉛直方向をy軸、y軸とz軸に直交する方向をx軸とする (図6)。なお、次節以降で行う画像の回転は、図中の赤い点を中心に行うものとする。この点は、炎の下端を目視で選択したものである。

### 5.1 平穏での揺らぎ

平穏での揺らぎは、閉め切った部屋などで見られるような、吹いている方向が特定できない程度の微小な空気の流れて生じる揺らぎを指す。図7は、実際に撮影した動画の一部である。平穏での揺らぎは、穏やかで、炎の乱れが生じないことが特徴である。また、稀に、縦方向の比較的大きな伸縮が生じるのも特

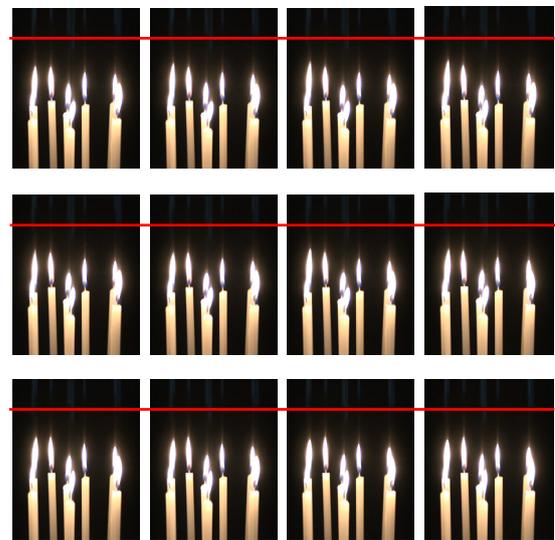


図7 実際に撮影した平穏での揺らぎ。  
縦方向の伸縮が生じたところの抜粋。

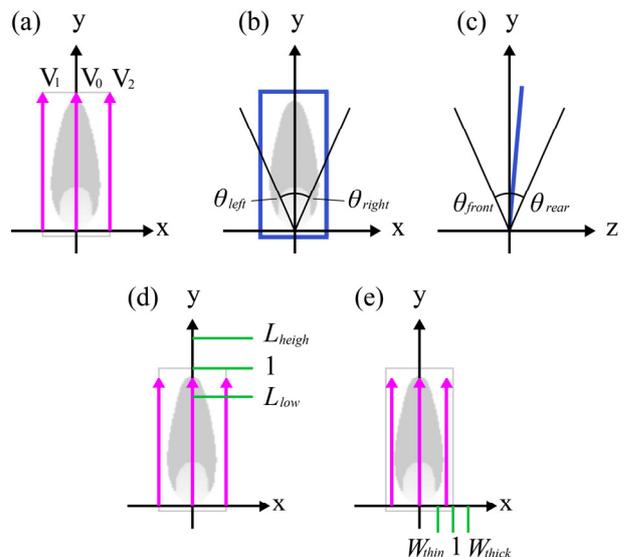


図8 平穏での揺らぎ

徴である。

この無指向性の揺らぎをノイズで表現する。時間効率を高めるために、スチルに対して、視覚的な効果性が高いと考えられる変形を組み合わせるアプローチをとる。具体的には、視点方向に対する、左右方向と前後方向の揺れ、および、縦方向と横方向の伸縮を組み合わせて、平穏での揺らぎを表現する。静止状態の炎を、視点方向に向けて配置したビルボードの状態であるとし、これに、揺らぎによる変形を加えていく。なお、左右の揺れ、縦方向の伸縮、および横方向の伸縮は、ビルボードの画像上の制御線の移動による画像の変形（モーフィング）であるのに対し、前後の揺れはビルボードそのものの回転を行っている点に注意されたい。

以下のような変形を行い、図8(a)の制御線 $V_0, V_1, V_2$ を移動させる。

**左右の揺れ:** 左右の揺らぎは、制御線の傾き角 $\theta_r$ を、最左の傾き角 $\theta_{left}$ および最右の傾き角 $\theta_{right}$ を用いて、 $\theta_r(t) = \text{lerp}(\theta_{left}, \theta_{right}, N_r(t))$ で求める(図8(b))。関数 $\text{lerp}$ は線形補間関数である。 $N_r(t)$ は正規化された $1/f^\beta$ ノイズである。この傾き角 $\theta_r$ を用いて、図8(a)の制御線 $V_0, V_1, V_2$ を図6の点を中心に回転させる。

**前後の揺れ:** 前後の揺らぎは、揺らぎ角 $\theta_f$ を、最前の傾き角 $\theta_{front}$ 、最背の傾き角 $\theta_{rear}$ 、および $1/f^\beta$ ノイズ $N_f(t)$ を用いて、 $\theta_f(t) = \text{lerp}(\theta_{front}, \theta_{rear}, N_f(t))$ で求める(図8(c))。なお、前後の揺らぎは、モーフィングではなく、ビルボードを回転させることで表現する。

**縦方向の伸縮:** 縦方向の伸縮率 $L_{hi}$ について、最伸と最縮時の伸縮率を表す $L_{high}, L_{low}$ および $1/f^\beta$ ノイズ $N_h(t)$ を用いて、 $L_{hi}(t) = \text{lerp}(L_{high}, L_{low}, N_h(t))$ で求める(図8(d))。伸縮率 $L_{hi}$ に応じて、制御線 $V_0, V_1, V_2$ を伸縮させる。

**横方向の伸縮:** 横方向の伸縮率 $W_u$ について、最厚と最薄時の伸縮率を表す $W_{thick}, W_{thin}$ および $1/f^\beta$ ノイズ $N_u(t)$ を用いて、 $W_u(t) = \text{lerp}(W_{thick}, W_{thin}, N_u(t))$ で求める(図8(e))。伸縮率 $W_u$ に応じて、制御線 $V_1, V_2$ の平行移動により、 $V_0$ から $V_1, V_2$ それぞれの距離を伸縮させる。

本節における変形は、画像の回転および拡大縮小のみで実現できるが、次節以降の揺らぎによる変換との組み合わせを行うために、制御線を用いている。本研究で使用したパラメタは表1のようである。

表1 平穏での揺らぎで用いたパラメタ

揺らぎ	ノイズの $\beta$ 値	パラメタ(1)	パラメタ(2)	パラメタの単位
左右の揺れ	2.8	$\theta_{left} : -5$	$\theta_{right} : 5$	度
前後の揺れ	2.8	$\theta_{front} : -15$	$\theta_{rear} : 15$	度
縦方向の伸縮	2.0	$L_{low} : 0.83$	$L_{high} : 1.00$	倍率
横方向の伸縮	2.8	$W_{thin} : 0.80$	$W_{thick} : 1.00$	倍率

## 5.2 軽風での揺らぎ

軽風は、平穏よりも強い風であり、ここでは、指向性を読み取る事のできる、そよかぜ程度の風を指す。図9は、実際に撮影した動画の一部である。観察の結果、軽風の中でも強弱による違いが存在し、風が強いほど炎の乱れが大きくなる傾向が認められた[13]。また、炎がろうそくの芯に張り付いてなびく現象が観察された(図9)。

本節では、軽風による指向性の揺らぎ生成について述べる。前節と同様、風はノイズで表現するものとし、時間効率を重視した揺らぎ表現のために、視覚的に効果性の高い変形を組み合わせるアプローチをとる。

基本的な揺らぎは、平穏での揺らぎと同様に、視点方向に対する、左右方向と前後方向の揺れ、および、縦方向と横方向の伸縮を組み合わせて表現する。これに、炎がろうそくの芯に張り付く表現(5.2.1節)、および、炎の乱れ(5.2.2節)の表現を加えて、最終的な揺らぎとする。静止状態の炎を、視点方向に向けて配置したビルボードの状態であるとし、これに、揺らぎによる変形を加える。なお、左右の揺れ、縦方向の伸縮、横方向の伸縮、芯にはりつく表現、および、炎の乱れの表現は、ビルボードの画像上の制御線の移動による画像の変形（モーフィング）であるのに対し、前後の揺れはビルボードそのものの回転を行っている点に注意されたい。

それぞれの変形は以下のものである。

**左右の揺れ:** 左右の揺らぎは、制御線の傾き角 $\theta_l$ を、横方向の

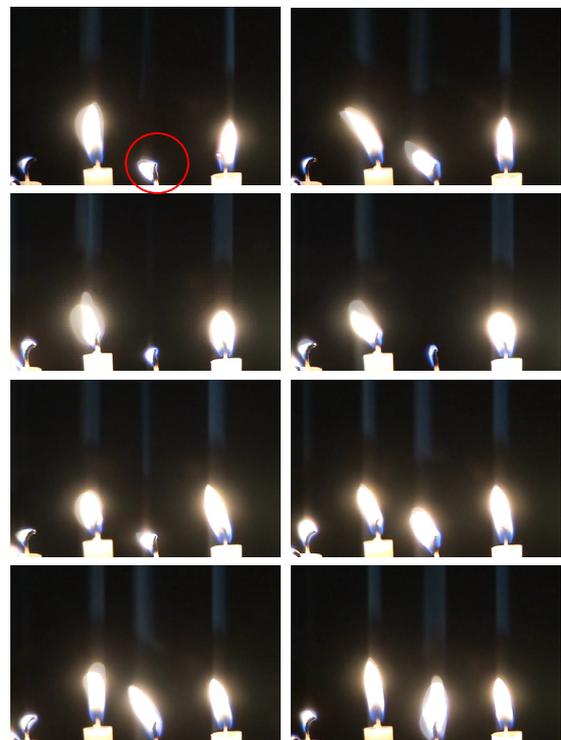


図9 実際に撮影した軽風での揺らぎ。炎がろうそくの芯に張り付いてなびく部分を抜粋。

最傾角  $\theta_{lean}$  および  $1/f^\beta$  ノイズ  $N_d(t)$  を用いて,  $\theta_l(t) = \text{lerp}(0, \theta_{lean}, N_d(t))$  で求める. 制御線の配置は, 5.1 節の左右の揺れと同様である. この揺れによる傾き角と, ろうそくの芯との角度がある一定値を超えた場合, 後述のような, ろうそくの芯に張り付いてなびく処理を行う.

**前後の揺れ:** 前後の揺らぎは, 揺らぎ角  $\theta_l$  を, 前後の最傾角  $\theta_{lean}$  および  $1/f^\beta$  ノイズ  $N_d(t)$  を用いて,  $\theta_l(t) = \text{lerp}(0, \theta_{lean}, N_d(t))$  で求める. 5.1 節の前後の揺れと同様に, モーフィングではなく, ビルボードの回転により表現する.

**縦方向の伸縮:** 5.1 節と同様である.

**横方向の伸縮:** 5.1 節と同様である.

本研究で使用した軽風のパラメタは表2のようなものである. なお, 一定値以上の風で, 炎を非表示とすることで, 消火も表現可能である.

表2 軽風での揺らぎで用いたパラメタ

揺らぎ	ノイズの $\beta$ 値	パラメタ(1)	パラメタ(2)	パラメタの単位
左右の揺れ	2.0-2.8	0	$\theta_{lean} : 60$	度
前後の揺れ	2.0-2.8	0	$\theta_{lean} : 20$	度
縦方向の伸縮	1.0-1.8	$L_{low} : 0.10$	$L_{high} : 1.00$	倍率
横方向の伸縮	2.0-2.8	$W_{thin} : 0.40$	$W_{thick} : 1.00$	倍率

### 5.2.1 炎がろうそくの芯に張り付く表現

軽風程度の揺らぎの場合, まれに炎がろうそくの芯に張り付いてなびく現象が観察される (図9). 本節ではこの表現の実現

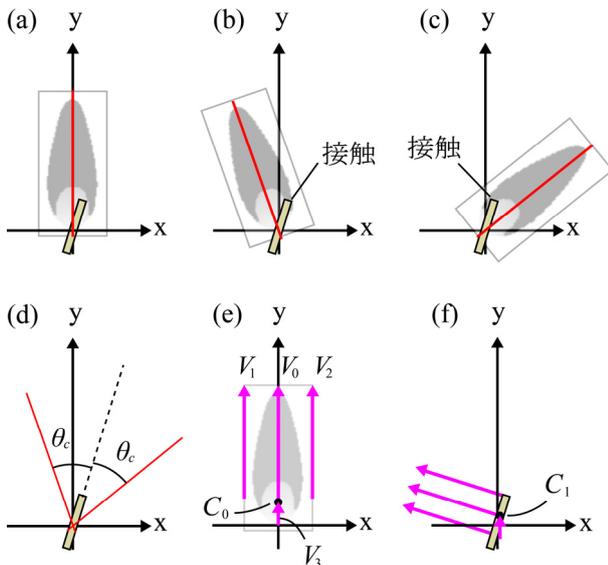


図10 炎がろうそくの芯に張り付く表現

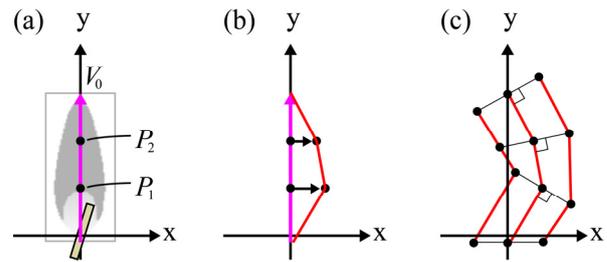


図11 炎の乱れの表現

手法を記述する. 観察の結果, 図10(b)や(c)のように炎の縁とろうそくの芯の先端とが接触することにより生じる傾向が認められた.

この観察から, 本手法では, 図10(d)のような左右の揺れに使用する制御線の傾き角  $\theta_l$  と, ろうそくの芯の角度の差が閾値  $\theta_c$  を超えた場合, 炎がろうそくの芯に張り付くものと判断する.

張り付いた場合には, 図10(e)のように4本の制御線を用いて, 炎がろうそくの芯に張り付いてなびく表現を行う. 制御線は, スチル上の点  $C_0$  (試行錯誤的に定めたスチル上の点) を用いて作成する. 制御線  $V_0, V_1, V_2$  で各制御線の根元を芯に張り付かせる. また, 制御線  $V_3$  の先端を, 図10(f)の点  $C_1$  (試行錯誤的に定めたろうそくの芯上の点) におく. 本論文では, 閾値は  $\theta_c = 35^\circ$  とした.

### 5.2.2 炎の乱れの表現

制御線の移動により炎の乱れを表現する. 炎の中心を通る, 縦方向の制御線  $V_0$  を  $n$  等分する. ここでは3等分として説明する (図11(a)). 実際, 計算時間との兼ね合いで, 3等分が適当である. 分割後の点を  $P_1(x_1, y_1), P_2(x_2, y_2)$  とする.

乱れの大きさを表す変位  $D$  を, 炎の最も乱れた時の変位  $D_{max}$  および  $1/f^\beta$  ノイズ  $N_d(t)$  を用いて,  $D(t, i) = \text{lerp}(-D_{max}, D_{max}, N_d(t-i))$  で求める.  $i$  は, 分割後の点  $P_i$  の添字 (例えば,  $P_1$  の場合は  $i=1$ ) である. すなわち,  $D$  は分割後の点を伝播していく.

変位  $D$  を用いて, 点  $P_1, P_2$  を移動させる. 移動は, 変形前の炎の傾きに応じて行う. 移動後の点をそれぞれ  $P_1'(x_1', y_1'), P_2'(x_2', y_2')$  とすると,

$$P_i' = (x_i + D(t, i), y_i) \quad (45 < \theta_r < 135) \quad (3)$$

$$P_i' = (x_i, y_i + D(t, i)) \quad (\theta_r \text{ が上記以外})$$

のように移動させる. ここで,  $\theta_r$  は制御線  $V_0$  と  $x$  軸とのなす角である.

他の揺らぎと組み合わせるために, 図11(c)のように, 制御線  $V_1$  と  $V_2$  も分割させる.  $V_1$  と  $V_2$  の分割は,  $V_0$  の分割で生じた制御線に直交する点を用いて分割する. これにより, 9本の制御線が生成される. なお, 5.2.1節の炎がろうそくの芯に張り付く処理も同時に行われる場合, 制御線は  $V_3$  を加えた10本となる.

### 5.2.3 軽風の生成

本章の冒頭で述べたように, ろうそくの炎は, 常に平穏での

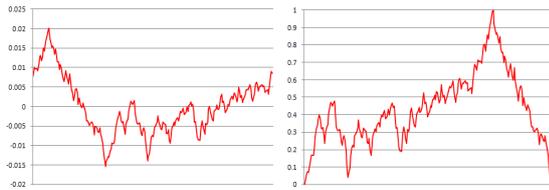


図 12 軽風の生成.

生成したノイズ (左) のシフトを行う (右).

揺らぎの影響下にあるものとしているために、平穏状態の風は、ノイズの周期性を利用して、ノイズのテーブルを繰り返し「再生」することで表現する。

一方、軽風は一時的に生じる揺らぎも表現できるように、軽風の発生時と抜けた時に、すなわち、 $N(0)=N(T)$ を0に近づけて、平穏の状態に自然に移行させる必要がある。Tは周期である。

本手法では、作成した  $1/f^\beta$  ノイズの最小値がテーブルの先頭に位置するように、作成したノイズの位相のシフトを行うことにより実現した (図 12)。なお、ノイズそのものの周期性は保証されているので、繰り返し再生することで、継続した風の表現は可能である。

### 5.2.4 複数のろうそくの揺らぎ

複数のろうそくに対して、軽風の指向性を表現するために、風の走りを考慮する。xz 平面状に配置された複数のろうそくの、自然な揺らぎを表現するために、ここでは、xz 平面に垂直な任意の面 S を風の発生源とし、風は平面 S の法線方向に吹くものとする。風の力は、面 S からの距離に応じて、遅延および減衰させて計算する。これにより生成した風は、5.2 節において、x 軸方向と z 軸方向とに分解し、左右および前後の揺れを表現する (図 13)。

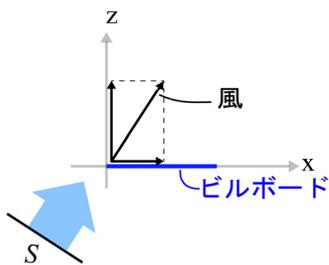


図 13 風の走り

### 5.2.5 平穏への収束までの動き

軽風が抜けた後に、ろうそくの炎は平穏の状態に移行するが、この移行の際に、振り子のように、ゆっくりと振動しながら減衰していくような様子が観察できる (図 9)。この平穏への収束までの動きを簡易的に再現することを試みる。

ここでは、左右の揺れと前後の揺れに対して、振り子のように、反対方向の揺れを与えることを繰り返し、減衰させる。左右の揺れで用いた、横方向の最傾角  $\theta_{lean}$  および前後の最傾角  $\theta_{lean}$

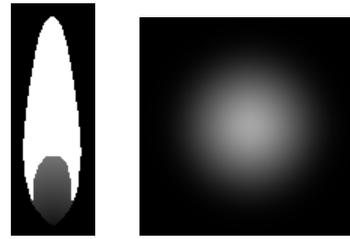


図 14 ろうそくの炎と炎が発する光.  
白が不透明、黒が透明を表す.



図 15 描画の順番

を用いて、減衰させて繰り返し適用させる。

## 6 レンダリング

第4章で、ろうそくと炎を、炎のスチル、炎が発する光、およびろうそくの3次元モデルの合成により表現することを述べた (図 5)。レンダリングは、それぞれの要素を、アルファブレンディングにより描画する。炎、および、炎が発する光のアルファ画像を図 14 に示す。描画は、炎が発する光、ろうそく、炎のスチルの順で行う。

ろうそくが複数あるシーンの場合、(1) すべてのろうそくの炎が発する光を描画し、その後、(2) 各ろうそくの本体と炎のスチルを描画すると、ろうそくが重なり合う場合でも自然な結果が得られる (図 15)。

炎が発する光は、試行錯誤的に、光の中心と炎の中心とが一致するように描画する。このとき、炎のサイズに応じて、光のサイズの変更を行う。ここでは簡単に、光の直径を、モーフィング後の  $V_1$  の始点と  $V_2$  の終点 (ビルボードの対角線) の長さに比例させる。また、視点からの距離に応じてアルファ値の減衰を行うことで、光の減衰を表現する。

## 7 実行結果

本手法で作成したアニメーション例を示す。図 16 に、モデルを1つ配置したときの、平穏での揺らぎのアニメーションを、そして、図 17, 18 に、軽風によるアニメーションを、それぞれ示す。図 19 は、モデルを複数配置したときのアニメーションで、平穏での揺らぎの例である。図 20 は、モデルを複数配置した時のアニメーションで、軽風での揺らぎの例である。使用したパラメータは、5.1 節の表 1, 5.2 節の表 2 に示す。

ユーザはインタラクティブに、任意のタイミングで軽風を与

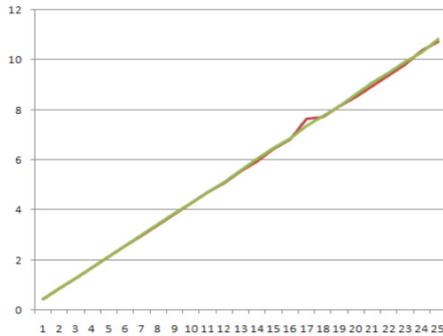


図 21 処理時間.

横軸はろうそくの本数, 縦軸は処理時間[ms]を表す.

赤色は平穏での揺らぎのみ, 緑色は平穏での揺らぎと軽風による揺らぎを行ったもの.

えることができる. 図 16-20 および付属のムービーに見られるように, 平穏や軽風での揺らぎは, 実際の炎と似た表現を生成できた. しかし, 実際の炎に軽風より強い風をあてたときに見られるような複雑な炎の動きは, 本手法による炎の変形とは異なる.

### 7.1 処理時間

本手法の処理時間の計測を行った. 実行環境は, OS: Windows XP, CPU: Pentium D 3.2[GHz], メモリ: 2[GB], ビデオカード: RADEON X600/X550 Series である.

図 21 および表 3 に, モデルの数を 1 から 25 までそれぞれ配置し, (1)平穏での揺らぎのみを行ったとき, および, (2)平穏での揺らぎと軽風による揺らぎを行ったとき, の実行時間の計測結果を示す.

(1)と(2)の差は小さく, 処理時間はモデルの数に大きく依存しているといえる. モデル数 25 の場合でも約 100[fps]で計算できるので, リアルタイムアニメーションが実現できたといえる.

表3 処理時間[ms]

ろうそくの数	(1)平穏での揺らぎ	(2)平穏での揺らぎと軽風による揺らぎ
1	0.437	0.435
5	2.14	2.15
10	4.27	4.28
15	6.41	6.50
20	8.53	8.64
25	10.7	10.8

## 8 まとめ

本論文では, 炎のスタイル (2次元の静止画像) に対してモーフィングを適用することで, 揺らぎのアニメーションを生成する手法を提案した. 空気の乱れを $1/f^\beta$ ノイズで表し, 炎の揺らぎ

をモーフィングの制御線の移動により表現した. そして, 本手法を実装し, リアルタイムに動作することを確認した.

今後の課題としては, 光の揺らぎや, 炎の色のちらつきの表現などによるリアリティの向上, 焚き火などの大規模炎への拡張などが挙げられる. また, 他現象についても, 現象面からのアプローチによる, リアルタイムで大規模な表現を可能とする技術の開発が課題として挙げられる.

### 謝辞

本研究の初期において, モーフィングによるアニメーション生成の可能性についての実験を試みて頂いた柳云善君 (現在, 富士通) に感謝します.

### 参考文献

[1] 竹下, 太田, 田村, 藤本, 村岡, 千葉. 2004. 爆発火炎の粒子ベースビジュアルシミュレーション法. 芸術科学会論文誌, 3, 2, 159-167.

[2] Takeshita, D., Ota, S., Tamura, M., Fujimoto, T., Muraoka, K., and Chiba, N. 2003. Particle-Based Visual Simulation of Explosive Flames. In Proceedings of Pacific Conference on Computer Graphics and Applications. 482-486.

[3] Nguyen, D., Fedkiw, R., and Jensen, H. 2002. Physically Based Modeling and Animation of Fire. In Proceedings of ACM SIGGRAPH 2002. 721-728.

[4] Ota, S., Tamura, M., Fujimoto, T., Muraoka, K. and Chiba, N. 2004. A Hybrid Method for Real-Time Animation of Trees Swaying in Wind Fields. The Visual Computer, 20, 10, 613-623.

[5] 宮内, 藤本, 村岡, 千葉. 2003. 線状物体の風による揺らぎの効率的なノイズベースアニメーション. 第 19 回 NICOGRAPH 論文コンテスト, 141-146.

[6] ソソラバラム, 藤本, 千葉. 2008. 平面状物体の風による揺らぎのノイズベースアニメーション法. 第 24 回 NICOGRAPH 論文コンテスト.

[7] 小川, 柳, 藤本, 千葉. 2008. 炎のスタイルからアニメーションを生成するノイズベースモーフィング. 第 24 回 NICOGRAPH 論文コンテスト.

[8] Schödl, A., Szeliski, R., Salesin, D. and Essa, I. 2000. Video Textures. In Proceedings of ACM SIGGRAPH 2000, 489-498.

[9] Okabe, M., Anjyo, K. and Onai, R. 2011. Creating Fluid Animation from a Single Image using Video Database. Computer Graphics Forum, vol. 30, no. 7 (Pacific Graphics 2011), pp. 1973-1982.

[10] 武者. 1994. ゆらぎの発想— $1/f$  ゆらぎの謎にせまる. 日本放送出版協会.

[11] Peitgen, H. and Saupe, D. 1988. The Science of Fractal Images. Springer.

[12] Beier, T. and Neely, S. 1992. Feature-Based Image Metamorphosis. In Proceedings of SIGGRAPH '92, 35-42.

[13] 新岡. 1994. 燃える—ろうそくからロケットの燃焼まで. オーム社.



図 16 平穏での揺らぎのアニメーション例



図 17 軽風での揺らぎのアニメーション例



図 18 軽風での揺らぎのアニメーション例. 制御線を表示.

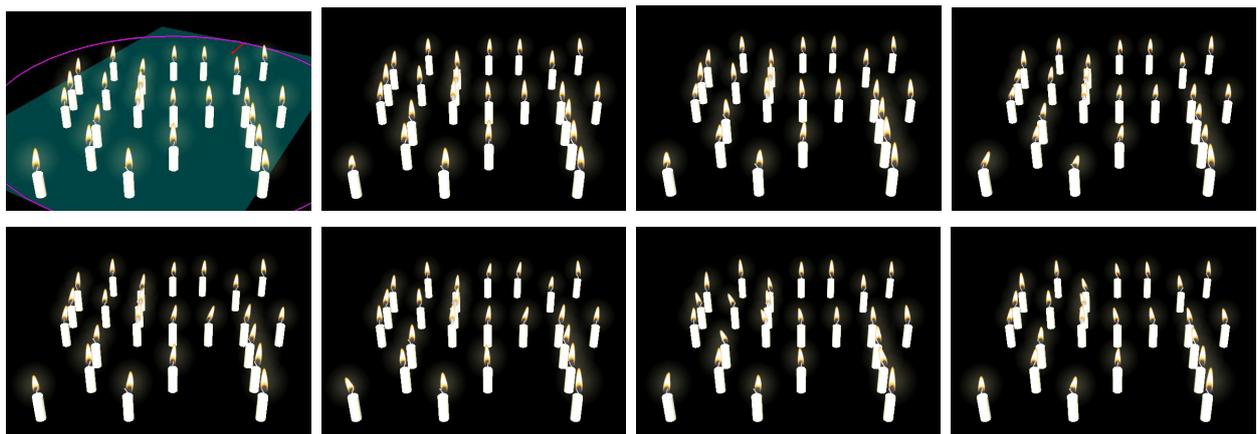


図 19 平穏での揺らぎのアニメーション例. 赤線が風の方角.

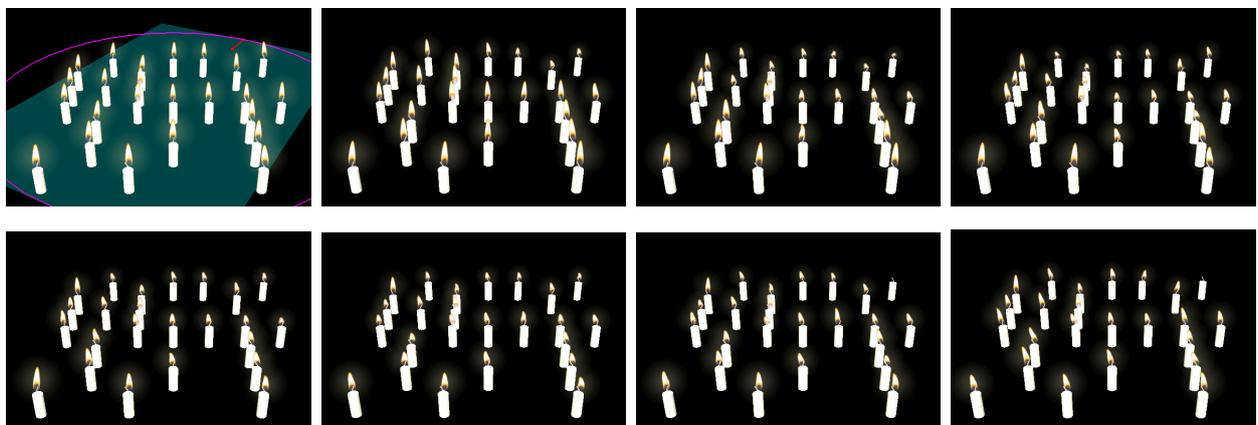


図 20 軽風での揺らぎのアニメーション例. 赤線が風の方角.