

## 滝のノイズベースアニメーション

高橋孝彰 藤本忠博 千葉則茂  
岩手大学

### アブストラクト

広大な滝のような大規模な流体现象を計算流体力学による数値シミュレーションによって表現するには、膨大な計算コストがかかってしまう。本研究では、滝の本流、飛沫、落水により発生する水煙、波といった滝の構成要素の総合的なアニメーションの生成法として、シンプルな粒子ベースの落水シミュレーションにノイズによる派生的な現象の表現を加えた効率的な手法を提案する。

#### 1. はじめに

これまで、炎や水の流体现象のCGによる表現法としては、特にここ10年間ほどは計算流体力学に基づく物理シミュレーションベースの方法が提案されてきている[1]。最近では小規模な現象においては非常にリアルな表現が可能となってきた[2][3][4][5]。一方、大規模な現象を表現するためには、シミュレーションベースの手法では計算時間の点で難点があるため、最近ではfBm (fractional Brownian motion) などのノイズを用いて、風やそれに伴う揺らぎ運動を表現する方法[6]や、2次元の流体シミュレーションの結果を3次元の流体现象の表現に拡張するハイブリッドな方法[7][8]が提案されてきている。また、[9]では、流体シミュレーションを行うことなく、流体のビデオ映像を3次元モデルにマッピングする手法により流体现象の拡張表現を実現している。

本研究では、滝の流れの概形を計算しておき、fBmを用いて軌道に沿って多数の粒子を配置し、データ増幅を行うというアプローチによる効率的な表現法を提案する。最近、印象的な碎波の表現法として、視覚的に非常に重要な派生的な現象である飛沫と泡の表現を行った方法が提案されている[10]。本研究での提案手法では、滝の本流、飛沫、水煙、および波のアニメーション生成を行っている。なお、本論文は[11]に基づき、シミュレーション法および、レンダリング法を拡張したものである。

#### 2. 滝の流れの概形の生成

本手法では、はじめに、滝の流れの概形をガイドラインと呼ぶ軌道の集合により表現する。また、障害物による流れの変化を実現するために、ガイドボックスと呼ぶボックス表現を用いる。その後、このガイドラインに沿って流れの実体をあらわす水の粒子を移動させ、さらに、ガイドボックスへの衝突判定により障害物による水のはね返りをあらわすような粒子を発生させることで、滝の流れを表現する。

#### 2.1 ガイドライン

滝の流れの概形、すなわち、流れのおおまかな全体形状を表現するため、ガイドライン粒子と呼ぶ粒子をいくつかを用意する。そして、これらの粒子の軌道(ガイドライン)を計算により求めることで、その軌道の集合により滝の流れの概形を表現する。なお、本研究では、 $x$ 軸方向、 $z$ 軸方向を水平方向とし、 $y$ 軸方向を鉛直方向としている(図1)。

各ガイドライン粒子の軌道の計算は、各粒子が鉛直方向の自由落下運動による放物運動をするものとして行った。図2に、これにより求めたガイドラインによる滝の流れの概形の例を示す。まず、3次元空間上でその粒子の運動の初期位置と終了位置を設定し、放物運動によってそれらの位置を粒子が通過するように、初期位置における初期速度を算出する。ここで、各ガイドライン粒子の初期速度の鉛直方向成分は自由に変更できるものとしている。そして、放物運動による時間経過ごとの粒子の位置を求める。

ガイドラインとして、以下の2種類を設定する。一つは、通常の流れを表現するための粒子生成ガイドラインである。もう一つは、障害物によって変化する流れを表現するための障害物ガイドラインである。後者は、流れの変化を実現するために、2.2節で述べるガイドボックスに変換し、流れを変えるための情報を格納しておく。

ガイドラインは、ひとまとまりの流れ、または、個々の障害物ごとにグループ化し、このグループをガイドライングループと呼ぶ。例えば、図2では、ガイドラインは3つのガイドライングループに分けられている。そして、3節で述べるように、ガイドラインに対して流れの実体をあらわす粒子(実体粒子)群をマッピングする際、ガイドライングループごとに実体粒子の生成を行う、また、ガイドライングループごとにガイドボックスへの進入判定を行うことにより、計算の高速化を図る。

## 2.2 ガイドボックス

本手法では、通常の流れを表現するためには2.1節で述べた粒子生成ガイドラインをそのまま用いるが、障害物ガイドラインについては、実体粒子と障害物との衝突を効率的に検出するため、障害物ガイドラインの形状をガイドボックスと呼ぶボックス表現に変換しておく(図3)。これは、実体粒子の衝突判定をガイドラインに対して直接行うより、ボックスに対する衝突判定のほうが容易に計算できるからである。3.2節で述べるように、実体粒子がガイドボックスに進入した場合、あたかも障害物に当たったかのような流れの変化を実現するため、実体粒子のガイドボックスへの進入判定を行い、ガイドボックスへの進入が検出された場合には、障害物でののはね返りをあらわす新たな実体粒子群を生成する。そして、この新たな各実体粒子に対して、進入粒子の進入位置と速度に基づく初期位置と初期速度を与え、はね返りによる運動を与える。なお、このとき、この新たな実体粒子の設定にガイドボックスを構成する障害物ガイドラインの情報が必要となるため、ガイドボックスには、障害物ガイドラインの番号を格納しておく。

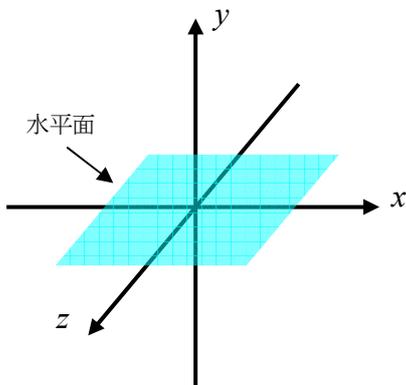


図1 3次元空間の設定

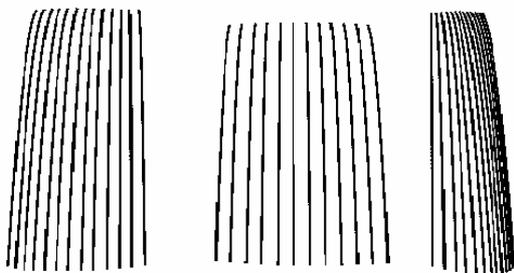


図2 ガイドラインによる滝の流れの概形

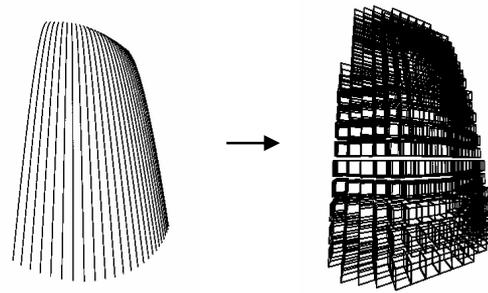


図3 ガイドボックス 左図：障害物ガイドライン 右図：左図をガイドボックスに変換したもの

## 3. 実体粒子による滝の流れの表現

2節の方法で生成したガイドラインおよびガイドボックスを用いて、滝の流れの実体を表現するための実体粒子群を適切に配置して運動を与えることにより、大規模な滝の流れを効率的に表現する。

なお、これ以降、特に断らない場合には、実体粒子を単に粒子と呼ぶことにする。

### 3.1 粒子生成ガイドラインへの実体粒子のマッピングによる流れの表現

2.1節で作成した粒子生成ガイドラインに対して、流れを表現する粒子群を適切な初期位置と初期速度を与えて生成し、ガイドラインに沿った適切な運動を与えることで流れを表現する。

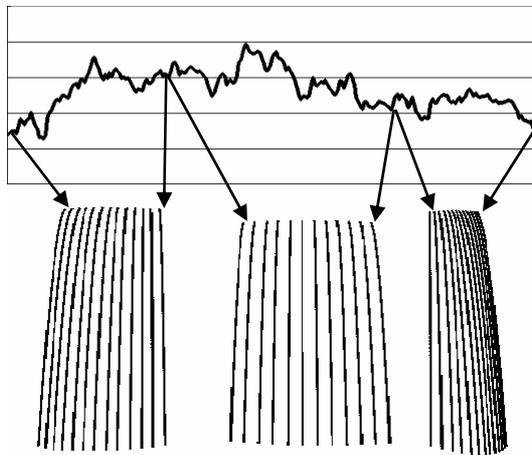
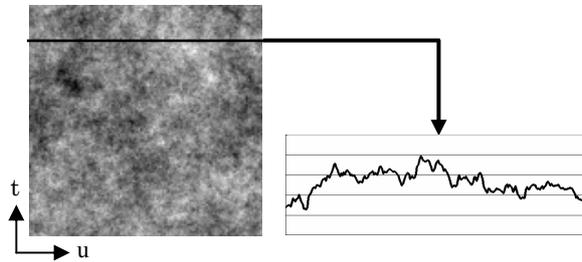
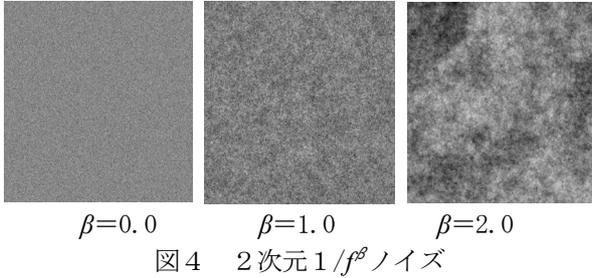
#### 3.1.1 粒子の生成

各粒子生成ガイドラインに対して、シミュレーションの時間ステップ $\Delta t$ ごとに、粒子群を生成する。各時刻 $t = n\Delta t$  ( $n$ は整数)で生成する粒子の個数 $P_{gen}$ は次式で与える。

$$P_{gen} = P_{gen\_weight} G_w^{F_{noise}(t,u)} \quad (1)$$

ここで、 $P_{gen\_weight}$ は粒子の生成数の調整係数であり、本流粒子か飛沫粒子かの粒子タイプにより異なる値とする。 $G_w$ は生成粒子係数であり、ガイドライングループごとに設定することにより各グループの生成粒子数をコントロールすることが可能である。また、 $F_{noise}(t,u)$ は2次元 $1/f^\beta$ ノイズ関数であり、時刻 $t$ と粒子生成位置 $u$ の関数として定義したものである。一般に、 $1/f^\beta$ ノイズとは、パワースペクトルが周波数の $\beta$ 乗に反比例しているノイズであり、図4に $\beta$ の値を変化させた2次元 $1/f^\beta$ ノイズの例を示す。本手法では図5に示すように時刻 $t$ における1列を1次元 $1/f^\beta$ ノイズとして取り出し、図6に示す

ように1次元  $1/f^\beta$  ノイズと粒子生成ガイドラインの位置を適用し、各ガイドラインに適用した位置のノイズの値から式(1)により生成する粒子の個数を決定する。



### 3. 1. 2 粒子の初期位置と初期速度の設定

3. 1. 1節の方法でガイドライン上に生成した粒子に対して運動を与えるため、その初期位置、ならびに、初期速度を以下のように設定する。

#### (1) 粒子の初期位置の設定

粒子の初期位置  $S_{init} = (S_{init_x}, S_{init_y}, S_{init_z})$  は次式で設定する。

$$Rn = Rand$$

$$S_{diff} = S_{weight} (1 - F_{noise}) Rand$$

$$S_{init_x} = Gs_x + Gn_x Rn + S_{diff} V_{init_x}$$

$$S_{init_y} = Gs_y + S_{y\_weight} F_{noise}$$

$$S_{init_z} = Gs_z + Gn_z Rn + S_{diff} V_{init_z}$$

(2)

式中の  $Gs = (Gs_x, Gs_y, Gs_z)$  は、対象としているガイドラインを求めた際のガイドライン粒子の初期位置(2. 1節参照)であり、この位置を基準に粒子の初期位置を与える。

水平方向位置  $S_{init_x}, S_{init_z}$  については、対象とするガイドラインとその隣のガイドラインとの間に粒子が置かれるようにする。具体的には、式(2)に示すように、対象とするガイドラインの初期位置から隣接ガイドラインの初期位置へ向かう3次元ベクトルの水平面(x-z平面)上での2次元成分  $Gn = (Gn_x, Gn_z)$  を用いて、その2つのガイドラインの初期位置を一樣乱数  $Rand$  ( $0.0 \leq Rand \leq 1.0$ ) による比率  $Rn$  で内分した位置を求める。そして、さらに、式(1)中のノイズ関数  $F_{noise}$  と一樣乱数  $Rand$ 、そして粒子の初期位置の調整を行うための係数  $S_{weight}$  を用いて定義される係数  $S_{diff}$  を用い、後述する式(3)で与えられる粒子の初期速度  $V_{init_x}, V_{init_z}$  と掛け合わせることで求めた変位分を加える。一方、鉛直方向位置  $S_{init_y}$  については、高さ調整係数  $S_{y\_weight}$  とノイズ関数  $F_{noise}$  の積により  $Gs_y$  からの変位を求める。

式(1)より、ノイズ関数  $F_{noise}$  の値が大きくなると粒子の生成数が多くなる。このとき、水平方向位置  $S_{init_x}, S_{init_z}$  については、係数  $S_{diff}$  の値が小さくなるため、粒子の初期位置が上記の2つのガイドラインの初期位置を結ぶ線分の周辺に集まるようになる。また、それと同時に、鉛直方向位置  $S_{init_y}$  の値は大きくなり、粒子の初期位置が高くなる。この結果、粒子の生成数が多くなるにつれて、水量が多く見えるような効果が生み出される。

#### (2) 粒子の初期速度の設定

粒子の初期速度  $V_{init} = (V_{init_x}, V_{init_y}, V_{init_z})$  は次式で設定する。

$$\theta = Rot_{gv} + Rot_p Rand_{pm}$$

$$V_{init_x} = Gv_x + V_{wh} Gv_{abs\_h} \sin \theta \quad (3)$$

$$V_{init_y} = Gv_y - V_{wv} Gv_y Rand$$

$$V_{init_z} = Gv_z + V_{wh} Gv_{abs\_h} \cos \theta$$

式中の  $Gv = (Gv_x, Gv_y, Gv_z)$  は、対象としているガイドラインを求めた際のガイドライン粒子の初期位置における初期速度(2. 1節参照)であり、この速

度を基準に粒子の初期速度を与える。

水平方向速度  $V_{init\_x}$ ,  $V_{init\_z}$  については、基準とする  $Gv_x$ ,  $Gv_z$  に対して、水平面 ( $x$ - $z$  平面) 上で図 7 に示すような扇形の範囲内で定義される角度  $\theta$  の方向を持つ変位分を与える。この角度  $\theta$  は、ガイドライン粒子の初期速度の方向  $Rot_{gv}$  を基準とし、時計・反時計回りの両方向に粒子速度方向係数  $Rot_p$  で決定される範囲内の方向を取る。式中の  $Rand$  は一様乱数、 $Rand_{pm}$  は負数も考慮した一様乱数 ( $-1.0 \leq Rand_{pm} \leq 1.0$ ) である。そして、ガイドライン粒子の初期速度の水平面上での 2 次元成分 ( $Gv_x$ ,  $Gv_z$ ) の大きさ  $Gv_{abs\_h}$  に水平方向の速度調整係数  $V_{wh}$  を掛け合わせた大きさを持ち、角度  $\theta$  の方向を向く 2 次元ベクトルを  $Gv_x$ ,  $Gv_z$  に対する変位分とする。

一方、鉛直方向速度  $V_{init\_y}$  については、基準とする  $Gv_y$  に対して、鉛直方向の速度調整係数  $V_{wv}$  と一様乱数  $Rand$  を用いて求める。なお、水平方向ならびに鉛直方向の速度調整係数  $V_{wh}$ ,  $V_{wv}$  には、本流粒子か飛沫粒子かの粒子タイプの違いにより異なった値を与える。

以上に述べたノイズ関数と一様乱数を用いた粒子の初期位置ならびに初期速度の設定法により、簡易的な方法で視覚的に自然な流れを実現することができる。また、特に、2 次元  $1/f^\beta$  ノイズの  $\beta$  の値を変更することで流れの激しさが調整でき、多様な流れや湧き出しの表現が可能である。

ガイドライン粒子の速度方向  $Rot_{gv}$

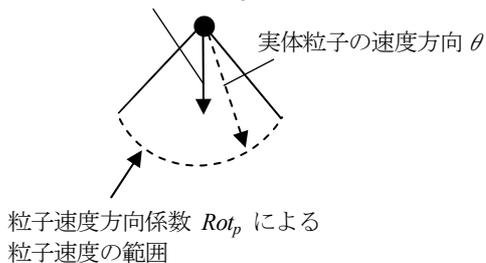


図 7 水平面 ( $x$ - $z$  平面) 上での粒子の初期速度方向の設定

### 3. 1. 3 粒子の運動シミュレーション

粒子タイプの違いにより、運動計算に違いを持たせることで、本流粒子、飛沫粒子のそれぞれに異なる運動を与え、自然な流れを表現する。

本流粒子の運動は、3. 1. 2 節で述べた初期位置と初期速度から開始し、時間ステップ  $\Delta t$  ごとに、時刻  $t = n\Delta t$  ( $n$  は整数) における粒子速度  $\mathbf{V}(n) = (V_x(n), V_y(n), V_z(n))$  を用いて式 (4) により粒子位置  $\mathbf{S}(n) = (S_x(n), S_y(n), S_z(n))$  を更新する。そして、粒子速度  $\mathbf{V}(n)$

を式 (5) で更新する。なお、水平方向の粒子速度  $V_x(n), V_z(n)$  は一定としている。

$$\begin{aligned} S_x(n+1) &= S_x(n) + V_x(n)V_{dh\_weight}(1+Rand)\Delta t \\ S_y(n+1) &= S_y(n) + V_y(n)\Delta t \\ S_z(n+1) &= S_z(n) + V_z(n)V_{dh\_weight}(1+Rand)\Delta t \end{aligned} \quad (4)$$

$$\begin{aligned} V_x(n+1) &= V_x(n) = V_{const\_x} \quad (\text{一定値}) \\ V_y(n+1) &= V_y(n) - G\Delta t \\ V_z(n+1) &= V_z(n) = V_{const\_z} \quad (\text{一定値}) \end{aligned} \quad (5)$$

同様に、飛沫粒子の運動は、3. 1. 2 節の初期位置と初期速度から開始し、式 (6) により粒子位置を更新する。粒子速度の更新には式 (5) を用いる。

$$\begin{aligned} S_x(n+1) &= S_x(n) + V_x(n)V_{fh\_weight}S_{move\_weight}(1+Rand)\Delta t \\ S_y(n+1) &= S_y(n) + V_y(n)S_{move\_weight}\Delta t \\ S_z(n+1) &= S_z(n) + V_z(n)V_{fh\_weight}S_{move\_weight}(1+Rand)\Delta t \end{aligned} \quad (6)$$

式 (5) 中の  $G$  は重力加速度係数、式 (4) と式 (6) 中の  $Rand$  は一様乱数である。また、式 (4) 中の  $V_{dh\_weight}$  は本流粒子の水平方向速度抑制係数であり、水平方向の粒子の揺らぎを表現する。同様に、式 (6) 中の  $V_{fh\_weight}$  は、飛沫粒子の水平方向速度抑制係数であり、 $V_{dh\_weight}$  に比べて大きな値を与えることで、飛沫粒子の水平方向のゆらぎが大きくなるようにしている。また、飛沫粒子の運動計算には、速度の絶対値を抑え、本流粒子よりもゆっくりと落下する運動を実現するために、 $S_{move\_weight}$  を飛沫粒子の移動量抑制係数として設定している。以上により、本流粒子の運動に比べ、より不規則に舞うような運動を飛沫粒子に与えている。

### 3. 2 ガイドボックスへの進入による粒子の生成

3. 1 節の方法で生成し運動を与えた本流粒子と飛沫粒子は、ガイドボックス (2. 2 節参照) に進入した場合には、障害物との衝突による流れの変化を表現するため、その進入位置で新たな粒子を発生させる。まず、運動している各粒子がいずれかのガイドボックスに進入するかどうかを、粒子とボックスの座標値による位置関係から求める。そして、粒子がボックスに進入した場合、進入位置のボックスに登録してあるデータから、進入した元の粒子のタイプ (本流粒子または飛沫粒子) と速度により、新たに発生させる粒子のタイプと初期速度を決定する。このとき、進入粒子の速度が大きな場合は、発生位

置と初期速度を変えた複数の粒子を発生させる。以下に、障害物との衝突により生成する粒子の設定方法を示す。

まず、新たに生成する粒子数  $P_{coll\_gen\_p}$  を以下の式により決定する。

$$P_{coll\_gen\_p} = 1 + \frac{2Coll_{gen\_p}|V_p|}{MAX\_V_d + MAX\_V_f} \quad (7)$$

ここで、 $V_p = (V_{p_x}, V_{p_y}, V_{p_z})$  は、衝突した元の粒子の速度であり、 $Coll_{gen\_p}$  は障害物との衝突により生成する粒子数を決定するための係数、 $MAX\_V_d$ 、 $MAX\_V_f$  は、それぞれ本流粒子、飛沫粒子の最大速度としてパラメタにより与えた値である。

生成する粒子のタイプを本流粒子と飛沫粒子のいずれにするかは、その生成確率をあらかじめ与えておくことにより、乱数により確率的に決定する。

生成する粒子の初期位置  $S_{coll} = (S_{coll\_x}, S_{coll\_y}, S_{coll\_z})$  は以下の式により決定する。

$$\begin{aligned} S_{coll\_x} &= S_{p\_x} + Coll_{s\_gen} Rand_{pm} \\ S_{coll\_y} &= S_{p\_y} + Coll_{s\_gen} Rand_{pm} \\ S_{coll\_z} &= S_{p\_z} + Coll_{s\_gen} Rand_{pm} \end{aligned} \quad (8)$$

ここで、 $S_p = (S_{p_x}, S_{p_y}, S_{p_z})$  は衝突した元の粒子の位置であり、 $Coll_{s\_gen}$  は粒子の生成範囲係数、 $Rand_{pm}$  は一様乱数である。

生成する粒子の初期速度  $V_{coll} = (V_{coll\_x}, V_{coll\_y}, V_{coll\_z})$  は以下の式により決定する。

$$\begin{aligned} \theta &= (Rot_{coll\_g} + Rot_{pv}) / 2 + Rot_{coll} Rand_{pm} \\ V_{coll\_x} &= Gv_{coll\_x} + \frac{V_{coll\_weight\_h}(V_{p\_x} + Gv_{coll\_x}) \sin \theta}{2} \\ V_{coll\_y} &= Gv_{coll\_y} - V_{coll\_weight\_v} V_{p\_y} Rand \\ V_{coll\_z} &= Gv_{coll\_z} + \frac{V_{coll\_weight\_h}(V_{p\_z} + Gv_{coll\_z}) \cos \theta}{2} \end{aligned} \quad (9)$$

式中の  $Gv_{coll} = (Gv_{coll\_x}, Gv_{coll\_y}, Gv_{coll\_z})$  は、衝突したボクセルを構成するガイドラインの速度であり、この速度を基準に粒子の初期速度を与える。

水平方向速度  $V_{coll\_x}$ 、 $V_{coll\_z}$  については、基準とする  $Gv_{coll\_x}$ 、 $Gv_{coll\_z}$  に対して、以下の変位分を与える。まず、水平面 (x-z 平面) 上での角度  $\theta$  として、衝突したボクセルを構成するガイドラインの速度方向  $Rot_{coll\_g}$  と衝突した元の粒子の速度方向  $Rot_{pv}$  を平均化した方向を基準とし、時計・反時計回りの両方向に速度方向係数  $Rot_{coll}$  で設定される範囲内に一様乱数  $Rand_{pm}$  で与えられる角度を考える。そして、衝突

した元の粒子の速度  $V_p$  と衝突したボクセルを構成するガイドラインの速度  $Gv_{coll}$  との平均速度の x, z 成分を考え、それに速度調整係数  $V_{coll\_weight\_h}$  を掛け合わせ、角度  $\theta$  の方向を持たせたものを変位分とする。一方、鉛直方向速度  $V_{coll\_y}$  については、基準とする  $Gv_{coll\_y}$  に対して、衝突した元の粒子の鉛直方向速度  $V_{p_y}$  に速度調整係数  $V_{coll\_weight\_v}$  と一様乱数  $Rand$  を掛け合わせたものを変位分とする。

以上の方法により新たに生成した粒子は、3. 1. 3 節で述べた方法で運動を与えられる。

### 3. 3 水煙粒子の生成と運動シミュレーション

本流粒子、あるいは、飛沫粒子が滝の下に位置する水面に着水した際、水煙粒子を生成する。その際の水煙粒子の生成位置と速度方向の決定方法を図 8 に示す。着水点を基準として、ある決められた範囲内に水煙粒子を生成する。また、水煙粒子の速度方向もその範囲内の方向とし、その速度は着水した粒子のタイプと速度により決定する。以下に、生成する水煙粒子の設定方法を示す。

生成する水煙粒子の粒子数  $P_{splash\_gen\_p}$  は以下の式により決定する。

$$P_{splash\_gen\_p} = Splash_{gen\_p} Fnoise_p \quad (10)$$

$Splash_{gen\_p}$  は、生成する水煙粒子数を決定するための係数であり、着水した粒子のタイプが本流粒子か飛沫粒子かにより異なる値とする。また、 $Fnoise_p$  は着水した粒子を生成する際に使用したノイズの値である。

生成する水煙粒子の初期位置  $S_{splash} = (S_{splash\_x}, S_{splash\_y}, S_{splash\_z})$  は以下の式により決定する。

$$\begin{aligned} \theta &= Rot_{pv} + Rot_{sp} Rand_{pm} \\ S_{splash\_x} &= S_{p\_x} + Splash_{s\_range} \sin \theta \\ S_{splash\_y} &= S_{p\_y} + Splash_{s\_v\_range} Rand \\ S_{splash\_z} &= S_{p\_z} + Splash_{s\_range} \cos \theta \end{aligned} \quad (11)$$

ここで、 $S_p = (S_{p_x}, S_{p_y}, S_{p_z})$  は、着水した粒子の着水点の位置である。角度  $\theta$  は水平面 (x-z 平面) 上で水煙粒子に与える速度方向を表し、図 8 に示すように、着水した粒子の速度方向  $Rot_{pv}$  を基準とし、一様乱数  $Rand_{pm}$  を用いて、時計・反時計回りの両方向に速度方向係数  $Rot_{sp}$  で設定される範囲内の値を取る。そして、この方向を向く大きさ  $Splash_{s\_range}$  の 2 次元ベクトルの x, z 成分を  $S_{p_x}$ 、 $S_{p_z}$  に加えたものを初期位置の水平成分  $S_{splash\_x}$ 、 $S_{splash\_z}$  とする。また、鉛直成

分  $S_{splash_y}$  は、着水点の鉛直成分  $S_{p_y}$  を基準とし、鉛直方向の位置を決定するためのパラメータ  $S_{splash_y\_range}$  と一様乱数  $Rand$  により決定する。

生成する水煙粒子の初期速度  $V_{splash} = (V_{splash_x}, V_{splash_y}, V_{splash_z})$  は以下の式により決定する。

$$\begin{aligned} \phi &= (\pi / 2) Rand \\ V_{splash\_x} &= |V_p| \cos \phi \sin \theta Rand \\ V_{splash\_y} &= |V_p| \sin \phi Rand \\ V_{splash\_z} &= |V_p| \cos \phi \cos \theta Rand \end{aligned} \quad (12)$$

水煙粒子の運動は、上記の初期位置と初期速度から開始し、時間ステップ  $\Delta t$  ごとに、時刻  $t = n\Delta t$  ( $n$  は整数) における粒子速度  $V(n) = (V_x(n), V_y(n), V_z(n))$  を用いて、式 (13) により粒子位置  $S(n) = (S_x(n), S_y(n), S_z(n))$  を更新する。なお、粒子速度  $V(n)$  は式 (5) で更新する。

$$\begin{aligned} S_x(n+1) &= S_x(n) + V_x(n)\Delta t \\ S_y(n+1) &= S_y(n) + V_y(n)Fnoise_{sp}(a, n)\Delta t \\ S_z(n+1) &= S_z(n) + V_z(n)\Delta t \end{aligned} \quad (13)$$

ここで、 $Fnoise_{sp}$  は水煙粒子に不規則な運動を与えるための2次元  $1/f^\beta$  ノイズ関数であり、 $a$  は水煙粒子ごとに乱数によって決定する値である。

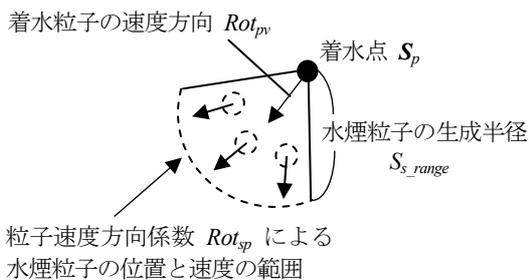


図8 水平面 ( $x-z$  平面) 上での水煙粒子の生成位置と初期速度方向の設定

### 3.4 波の運動の生成

滝の落ちる水面に2次元セルを設置し、各セルに進入した粒子により、セルの高さを変化させ波を発生させる。また、波の運動は離散化した波動方程式を解くことにより計算する[12][13]。シミュレーションの  $n$  ステップ目における位置  $(i, j)$  のセルの高さを  $h_{ij}^n$  とすると、 $n-1$  および  $n$  ステップ目で求められたセルの高さを  $h_{ij}^{n-1}$  とし、 $n+1$  ステップ目のセルの高さを式 (14) で求める。

$$h_{ij}^{n+1} = \frac{c^2 \Delta t^2}{d^2} (h_{i+1,j}^n + h_{i-1,j}^n + h_{i,j+1}^n + h_{i,j-1}^n) + (2 - \frac{4c^2 \Delta t^2}{d^2}) h_{ij}^n - h_{ij}^{n-1} \quad (14)$$

ここで、 $c$  は波の速度 (全セルにおいて一定)、 $d$  はセルの一辺の長さ、 $\Delta t$  はシミュレーションの時間ステップである。

## 4. レンダリング方法

粒子群により表現された流体の運動をレンダリングするための方法としては、**Marching Cubes** 法[14]や四面体格子法[15]により水面をポリゴン化してレンダリングする方法や、粒子を直接レンダリングするポイントベースレンダリング法[16][17]がある。本研究では、後者の方法を基に粒子群のレンダリングを行い、水面の波や地形については、ポリゴンベースのレンダリングを行う。

### 4.1 粒子のレンダリング

本流粒子と飛沫粒子は、時間ステップ間での移動量が大きく、通常のポイントベースレンダリングにより粒子をそのまま点として描いたのでは不自然な映像となってしまう。そこで、その時点における粒子の位置と速度から、時間ステップ  $\Delta t$  間の移動による軌道を計算し、その軌道に合わせてテクスチャをマッピングするという方法により、本流粒子と飛沫粒子のレンダリングを行う。その際、図9(a)に示す  $\alpha$  値の分布テクスチャを用いた。このテクスチャは、今回はペイントソフトを用いて作成したものであるが、ガウス分布を利用して作成するなど、任意のものとする事ができる。このテクスチャを粒子の軌道にマッピングする際には、その縦方向の長さについては、粒子の軌道の長さ ( $\Delta t$  間の移動距離) を基本長として、あらかじめ与えた任意の倍率で伸縮させて用いた。また、横方向の幅もあらかじめ与えた任意の倍率で伸縮させて用いた。

一方、水煙粒子は、本流粒子と飛沫粒子に比べて時間ステップ間での移動量が小さいため、図9(b)の円形のテクスチャを粒子の位置にマッピングする。このテクスチャもペイントソフトを用いて作成したものであるが、ガウス分布など、任意のものとする事ができる。また、このテクスチャの大きさは水煙粒子を生成する際に乱数を用いて決定する。

各粒子に対して上記のテクスチャを用いて、粒子群全体を光の減衰を考慮して陰影を表現するZバッファベースのシェーディング法[18]によりレンダリングを行った。この手法により、光源および粒子間の前後関係を考慮した陰影付けを行なうことで、流れの奥行き感を表現することができる。

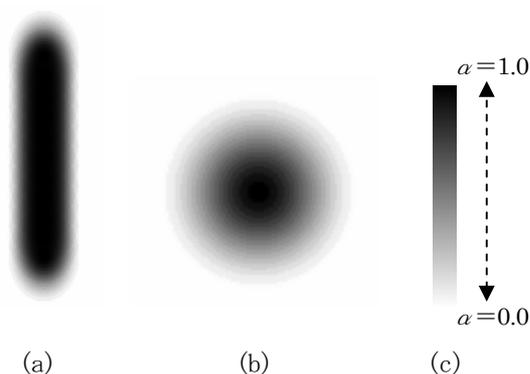


図9  $\alpha$  値の分布テクスチャ (a) 本流粒子と飛沫粒子用テクスチャ (b) 水煙粒子用テクスチャ (c) 図中の輝度(明暗)と  $\alpha$  値の関係

#### 4.2 波のレンダリング

水面に発生する波は、セルの位置と高さを用いて三角形パッチを生成し、水面のテクスチャを用いてレンダリングを行う。

#### 4.3 背景の表現

滝の背後の地形を表すために、ガイドラインのグループごとに、ガイドライン間にポリгонを張り、岩盤のテクスチャをマッピングする。また、ガイドラインのグループ間の隙間は、単純に一枚のポリゴンで埋め、岩盤のテクスチャをマッピングする。

また、空や前景についても、テクスチャマッピングを用いた。

#### 5. 実行結果

障害物の有無と粒子数を変更していくつかのケースについて、提案手法を用いて滝のアニメーション生成を行った結果を表1に示す。表中の総粒子数は、シミュレーション中で流れが視覚的に安定した状態のある時刻について、シミュレーション空間中に存在する粒子数を計測したものである。また、シミュレーション時間ならびにレンダリング時間は、その時刻における全粒子に対する1ステップの計算に要した時間である。シミュレーション時間は3節の方法で粒子の運動を計算するのに要した時間、レンダリング時間は4節の方法でレンダリングを行うのに要した時間である。膨大な粒子数に対して、粒子の運動が効率的に求められていることがわかる。また、粒子数の増加に対して、シミュレーション時間がほぼ比例して増加している。

図10、図11は、表1(b)、(d)のアニメーション画像である。図中の各画像はアニメーションのフレームを抜き出したものであり、上から下に向かう順で時間が経過している。最上段の画像は、背景(地形、水面、空)のみをレンダリングしたもので、丸で囲んだ位置に障害物を設定している。これらのアニメーション画像から、4.1節のレンダリング法による粒子の色の陰影付けにより、流れの奥行き感が表現されていることがわかる。また、3.2節で

述べたガイドボックスを用いた方法により、粒子が障害物に衝突した際の流れの変化が実現されている。

表1 シミュレーションの計算時間の比較

	総粒子数	障害物	シミュレーション時間	レンダリング時間
a	137,721 個	無	0.10 秒	0.58 秒
b	162,133 個	有	0.15 秒	0.71 秒
c	582,349 個	無	0.42 秒	1.73 秒
d	611,593 個	有	0.49 秒	2.21 秒

表2は、本手法の計算効率を示すため、本手法とMPS法による3次元シミュレーションに関する計算時間の比較を示したものである。ここで、MPS法とは、計算流体力学の手法の一つであり、流体を粒子の集合として表現し、粒子間の相互作用を計算し流体の運動を求める手法である。今回の実験では、表2に示すように、ほぼ同じくらいの粒子数のケースで、MPS法は本手法より2桁程度多くの計算時間を要した。また、計算機上でのプログラムの実行について、粒子の運動計算に簡易的な計算式を用いる本手法に対して、MPS法では粒子数に応じた大規模な連立方程式を解く必要があり、大きなメモリ領域を要する。ここで、実際のアニメーション生成について考えた場合、図10、図11に示す規模の滝のアニメーションを生成するためには、表1に示した程度の粒子数が必要とされる。そして、表2から、仮にMPS法でこの程度の粒子数を扱おうとする場合、膨大な計算時間を要することが予想される。さらに、使用するメモリ領域も膨大なものとなり、プログラムの実装方法にもよるが、実際には必要なメモリ領域を確保できない可能性もある。また、特に飛沫や水煙を伴う滝の流れを考えた場合、通常のMPS法による粒子運動として計算できるのは本流のみであり、飛沫や水煙まで流体計算で求めるためには、本流に用いる粒子よりさらにサイズの小さな粒子を多数使い、空気粒子なども考慮しなければならない。実際、これまで、図10、図11に示す程度の規模の飛沫や水煙を伴う流体運動を3次元のMPS法で計算してアニメーションを生成した事例はなく、現実的に、MPS法等の計算流体力学による数値シミュレーションですべての計算を行うこと自体、実施が困難であり、得策ではない。一方、本手法は、映像生成のための実用性を目指したものであり、出来る限りリアルタイムに近い時間で映像を生成し、試行錯誤を繰り返しながら目的とする映像を作り込んでいくことを可能とすることを目標とする。そして、実験の結果からも、実際の映像生成への利用を考えた場合、計算時間の点に関する本手法の有効性がわかる。

なお、上記のシミュレーションで用いた2次元  $1/\beta^2$  ノイズを格納する配列のサイズは  $1024 \times 1024$ ,  $\beta$

の値は 1.50, 時間ステップ  $\Delta t$  は 0.03 としている。使用した計算機は, Pentium4 2.6GHz, メモリ 1024 MB である。

表2 MPS法と本手法によるシミュレーションの計算時間の比較

	総粒子数	シミュレーション時間
MPS法	20,654 個	6.97 秒
本手法	22,155 個	0.015 秒

## 6. おわりに

本研究では, 滝の流れにおいて特徴的である本流, 飛沫, 水煙, 波を効率的に表現する滝のアニメーション生成法を提案した。本手法によれば, 通常の計算流体力学による方法では膨大な計算コストを要する大規模な滝の流れでも, シンプルな方法で効率的に表現することが可能である。

実験の結果, 計算流体力学の手法の一つであるMPS法によるシミュレーションと比べ, 1ステップの計算時間が大きく短縮されている。また, 障害物の有無で計算時間に大きな差がなく, 粒子と障害物との衝突のための計算コストが, 他の部分の計算に比べ小さいことがわかる。さらに, 異なる粒子数によるシミュレーション時間を比較すると, 粒子数の増加に対してほぼ線形に計算時間が増加していることが分かり, 本手法の効率性が確認できた。しかしながら, 計算流体力学による手法とは異なり, 粒子同士の相互関係等を考慮に入れていないため, 粒子の運動に不自然な部分が見られる。粒子同士の影響を考慮に入れた運動計算を行うことで, 例えば, 粒子間の引力や斥力などによる相互作用により, よりリアルな厚みある流れが表現されるなど, より良い結果が得られる可能性があり, これらの改良が今後の課題である。

レンダリングについては, 陰影計算を行った結果, 粒子同士の影響により光の減衰が表現され, 流れの奥行き感があらわれている。しかしながら, より品質の高い映像を得るためには, 周囲の粒子との関係を考慮に入れ, 液体の表面としてレンダリングする方法や, 周囲環境の映り込みが表現可能であるレンダリング法の開発などが今後の課題である。

## 参考文献

[1] N.Chiba, S.Sanakanishi, K.yokoyama, I.Ootawara, K.Muraoka and N.Saito, "Visual Simulation of Water Currents Using a Particle-based Behavioural Model", The Journal of Visualization and Computer Animation, Vol.6, No.3, pp.155-171, 1995.  
 [2] N.Foster and D.Metaxas, "Modeling the Motion of a Hot, Turbulent Gas", SIGGRAPH 97, pp.181-188, 1997.  
 [3] R.Fedkiw, J.Stam and H.Jensen, "Visual

Simulation of Smoke", SIGGRAPH 2001, pp.15-22, 2001.

[4] D.Enright, S.Marschner and R.Fedkiw, "Animation and Rendering of Complex Water Surfaces", SIGGRAPH 2002, pp.736-744, 2002.

[5] Nick Foster and Ronald Fedkiw, "Practical Animation of Liquids", SIGGRAPH 2001, pp.23-30, 2001.

[6] 宮内俊輔, 藤本忠博, 村岡一信, 千葉則茂, "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, pp.141-146, 2003.

[7] Nick Rasmussen, Duc Quang Nguren, Willi Geiger and Ronald Fedkiw, "Smoke Simulation For Large Scale Phenomena", SIGGRAPH 2003, pp.703-707, 2003.

[8] 鈴木亨, 藤本忠博, 村岡一信, 千葉則茂, "2次元物理シミュレーションに基づく効率的な3次元砕け波のアニメーション生成法", 第19回 NICOGRAPH 論文コンテスト, pp.135-140, 2003.

[9] Kiran S. Bhat, Steven M. Seitz, Jessica K. Hodgins, Pradeep K. Khosla, "Flow-based Video Synthesis and Editing", SIGGRAPH 2004.

[10] Tsunemi Takahashi, Hitroto Fujii, Atsushi Kunimatsu, Kazuhiro Hiwada, Takahiro Saito, Ken Tanaka and Heihachi Ueki, "Realistic Animation of Fluid with Splash and Form", EUROGRAPHICS 2003, pp.391-400, 2003.

[11] 高橋孝彰, 藤本忠博, 千葉則茂, "滝のノイズベースアニメーション", 第20回 NICOGRAPH 論文コンテスト, pp.29-34, 2004.

[12] "GAME PROGRAMING GEMS", 株式会社ボーナデジタル, pp.180-187, 2001.

[13] Trim.D.W, Applied Partial Differential Equations, PWS-Kent, 1990.

[14] William.E.Lorensen and Harvey.E.Cline, "Marching Cubes : A High Resolution 3D Surface Construction Algorithm", SIGGRAPH 1987, vol.21, Number4, pp.163-169, 1987.

[15] Bernardo Piquet Carneiro, Claudio Silva e Arie Kaufman, "Tetra-Cubes : An Algorithm to Generate 3D Isosurfaces Based upon Tetrahedra", 1996.

[16] Hanspeter Pfister, Matthias Xwicker, Jeroen van Baar and Markus Gross, "Surfels : Surface Elements as Rendering Primitives", SIGGRAPH 2000, pp.335-342, 2000.

[17] Szymon Rusinkiewicz and Marc Levoy, "Qsplat : A Multiresolution Point Rendering System for Large Meshes", SIGGRAPH 2000, pp.343-352, 2000.

[18] 太田真, 田村真智子, 藤田邦彦, 藤本忠博, 村岡一信, 千葉則茂, "天候景観のためのZ-バッファベース高速3Dテクスチャレンダリング法", 平成14年度電気関係学会東北支部連合大会, p.336, 2002.

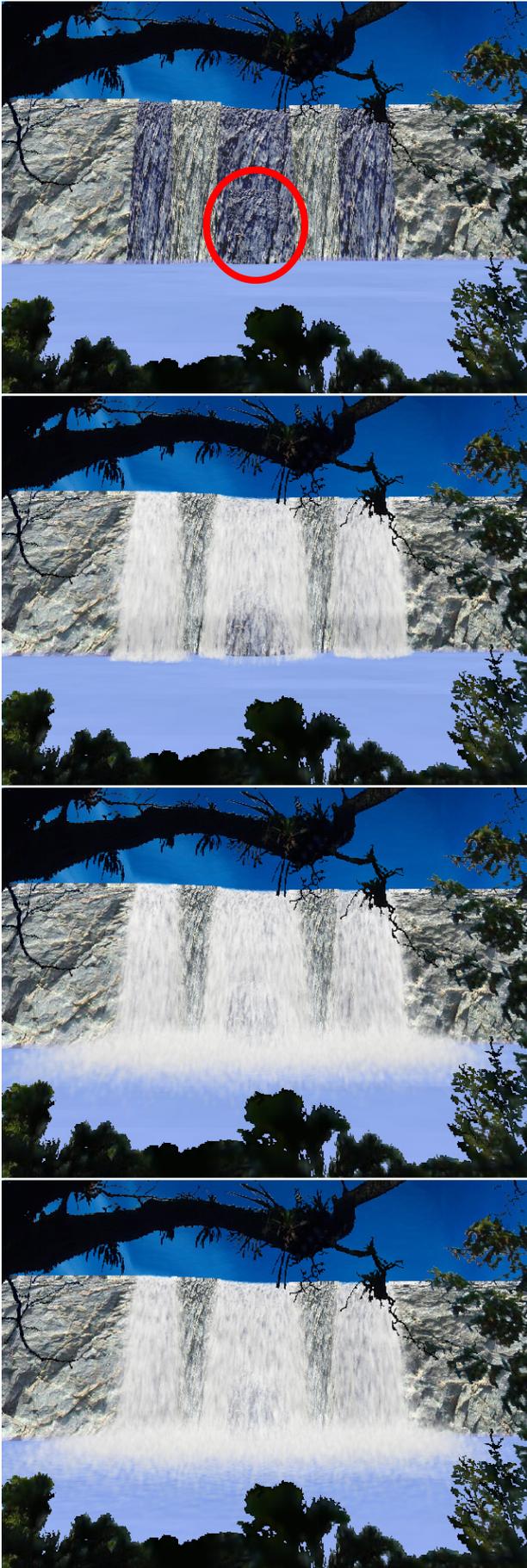


図10 表1(b)のシミュレーション結果



図11 表1(d)のシミュレーション結果