

## 粒子法による雨染みのシミュレーションの GPU による高速化

田中将貴<sup>1)\*</sup> (非会員)

床井浩平<sup>2)</sup> (正会員)

1) 和歌山大学システム工学研究科

2) 和歌山大学システム工学部

## GPU Acceleration of Particle Method Simulation of Rain Stains

Masaki Tanaka<sup>1)</sup> (Non-Member)

Kohe Tokoi<sup>2)</sup> (Member)

1) Graduate School of Systems Engineering, Wakayama University

2) Faculty of Systems Engineering, Wakayama University

<sup>2)</sup> tokoi@sys.wakayama-u.ac.jp

### アブストラクト

コンピュータグラフィックスにおいてオブジェクトの外観の経年変化を表現するために、オブジェクト表面上に雨染みを再現する手法を提案する。これには粒子法の一手法である SPH (Smoothed Particle Hydrodynamics) 法によるシミュレーションを用いることができる。本研究では SPH 法を拡張し、水の付着力の影響を考慮した雨染みの形状を生成する手法を開発した。提案手法は水滴が物体表面を伝う現象や水滴が滴る現象を再現するため、従来手法より現実に近い汚れの模様を生成できる。また本研究では、提案手法を GPU 上に実装し、シミュレーションに用いる粒子の各種の物理量を GPU のメモリ上に配置して、処理の高速化を図った。

### Abstract

This paper describes a method to reproduce rain stains on the surface of aging objects in computer graphics. We extended the SPH (Smoothed Particle Hydrodynamics) method and generated the pattern of rain stains that take the influence of adhesive force of water into account with it. Since the proposed method simulates the behavior of water droplets along the object surface and the behavior of water dripping, it can generate more realistic patterns of the rain stains than conventional methods does. In addition, we implemented the proposed method on GPU, and accelerated the generation of rain stain generation by placing various physical quantities of the particles on the memory of the GPU.

---

\* 現在, iCAD 株式会社  
Presently iCAD Limited.

## 1. はじめに

### 1.1. 研究背景

自然環境にさらされている物体表面の状態は、環境の影響を受けて時間とともに変化する。この原因には風雨の浸食や錆、汚れの付着などがある。コンピュータグラフィックスにより現実感のある映像を作成するには、このような経時変化を再現する必要がある。中でも雨染み（雨による汚れ）の模様は物体形状にも依存し、複雑なものになる。

従来このような雨染みの再現には、物体に手作業で汚れを描き込む手法が一般的に用いられてきた。しかし、これは一般に多くの手間と時間を要する。そこで、雨染みを自動的に生成する手法が提案されている[1][2]。また、雨の影響の再現には流体シミュレーションが用いられるが、これは計算コストが高いため、GPU の利用などさまざまな高速化手法が検討されている[3][4][5]。本研究はこれを雨染みの生成の高速化にも応用する。

### 1.2. 関連研究

Dorsey らは、水と物体表面の相互作用を微分方程式でモデル化し、これに SPH (Smoothed Particle Hydrodynamics) 法を組み合わせて、自然で複雑な汚れを生成する手法を提案している[1]。

Chen らは経年変化の再現に、陰影付けにおける大域照明の手法を応用した、 $\gamma$ -ton tracing 法を提案している[2]。この手法は経年変化の原因となる場所から放出した、汚れの原因となる粒子の挙動を追跡し、その結果を格納した  $\gamma$ -ton map を用いて汚れの量や劣化の程度を決定する。

これらの手法は雨染みの発生メカニズムの再現を行うものであり、生成された汚れは、いずれも十分自然に見える。しかし、これらは雨滴が物体表面を伝う現象や、水滴が滴る現象による汚れの洗浄や沈着などの再現が十分ではない。

内田らは氷から溶け出した水が滴る現象や、氷表面が水が伝う現象を再現するために、SPH 法に表面張力と界面張力を加えたモデルを提案した[6]。これにより、SPH 法だけでは困難だった、物体表面上での水の挙動の再現が可能になった。

そこで本稿では、雨染みの模様を生成するために、内田らのモデルを氷以外の任意の材質への付着力を考慮するよう拡張した。さらに、付着力を壁重み関数にもとづいてモデル化することにより、壁粒子を用いずに付着力の計算を可能にし、計算速度の向上を実現した。

### 1.3. 研究目的

前述のように、本研究では水滴の付着力の影響を反映した雨染み模様の生成を目的とした。この付着力を決定する仮想的なパラメータとして、本研究では界面張力係数を導入した[12]。これにより、雨染みを水滴の付着力の違いによって描画し分けることが可能になった。

また、水滴の付着力を壁重み関数化して計算量の削減を図った。以上をもとに、雨染みの生成を行う手法を開発した。また、計算コストの高い流体シミュレーションの高速化を図るため、提案手法を GPU 上に実装した。

## 2. 雨染み

雨染み模様は主として雨滴が物体表面を流れることよって発生する。この形状が一般に複雑なのは、雨滴の流れの主要な決定要因が重力であるものの、このほかに物体表面の材質や汚れの素材による付着力、風や温度の影響など多数の要因が関わるためであると考えられる。しかし、これらのうち雨染みの模様を決定する主たる要因は、物体形状のほか、雨滴とその物体表面との間に働く付着力である。本章では、この付着力に着目した雨染みの形状の生成手法について説明する。

本研究では、雨による汚れの発生要因として、洗浄と堆積の二つを考慮した。洗浄は物体表面に付着した汚れを雨滴が吸着し洗い流す作用であり、堆積は洗浄などにより雨滴に含まれた汚れの成分が水分の蒸発によりその場に滞留する作用である。

自然環境下では様々な汚れが存在し、汚れが物体表面に付着する要因も非常に多様である[7]。したがって、これらを正確にモデル化することは難しい。そのため本稿では、汚れを汚れの水溶性により分類した。橋高らは物体の汚れの付き易さと物体の濡れ易さに関係があるとしている[8]。これは、物体が濡れやすいと水溶性の汚れが付き易く、反対に物体が濡れにくいと非水溶性の汚れが付き易いということである。本稿でもこの考えを採用し、これもとに汚れを分類して、それぞれの性質と発生メカニズムについて説明する。

水溶性の汚れは、水に溶けた汚れが、親水性の高い物体表面に定着あるいは水とともに内部に沈着することによって発生する。このため、この汚れは水が流れる部分に発生する。一方、非水溶性の汚れは、水に溶けにくい汚れが、水が滞留する部分に定着することにより発生する。この汚れは水が流れる部分では洗浄されやすく定着しない。図 1 に雨染みの例を示す。水溶性の汚れと非水溶性の汚れには次のものがある。

- 水溶性の汚れ：錆汁（図 1 左）、泥、埃
- 非水溶性の汚れ：砂、排気ガスや煤煙（図 1 右）



図 1 雨染みの例

## 3. 提案手法

提案手法は最初に乱数にもとづいて水の粒子を生成し、重力方向に不規則かつ均一に降り注ぐ雨を再現する。そして界面張力による物体表面への付着力を導入した SPH 法を用いて、物体表面に降り注いだ雨滴の挙動を再現する。この付着力の大きさは界面張力係数によって制御する。この雨滴と物体表面との衝突の際に、水の粒子と物体表面との間で汚れの移流が生じる

と考えることにより、雨染みの模様の発生をシミュレートする。また、これに格子を用いた近傍粒子探索と壁重み関数による壁境界条件の事前計算を導入して、処理の高速化を行う。

なお、現実世界では、雨には外力として重力以外に空気の流れや空気抵抗など様々な力が働く。これらは現実では雨染みの形に影響を与えるが、本研究ではこれらの影響を考慮しない。

### 3.1. 雨のモデル化

本研究では、流体の挙動を表現する手法として、粒子法の一手法である SPH 法[9]を用いる。SPH 法はポワソン方程式を解かない。そのため、他の粒子法の手法と比べアルゴリズムが比較的単純であり、リアルタイムシミュレーションなど処理速度が重要である場合に向いている。

粒子  $i$  の速度は、以下のナビエ・ストークス方程式 (1) により求められる。ここで  $\rho$  は密度、 $D/Dt$  はラグランジュ微分 (流れの上での物理量の変化)、 $\mathbf{v}_i$  は速度、 $p$  は圧力、 $\mu$  は粘性係数、 $\mathbf{f}$  は外力を表す。ここで想定している外力は重力である。

$$\rho_i \frac{D\mathbf{v}_i}{Dt} = -\nabla p_i + \mu \nabla^2 \mathbf{v}_i + \mathbf{f} \quad (1)$$

この右辺はそれぞれ圧力項、粘性項、外力項である。SPH 法により流体シミュレーションを行うには、各タイムステップで一つ一つの粒子に対して上式を解き、粒子の速度から粒子位置を求めることにより流体をシミュレートする。このナビエ・ストークス方程式は Müller らの手法[10]を用いて定式化される。

### 3.2. 付着力のモデル化

#### 3.2.1. 液体の付着力

液体が物体表面を伝う現象や滴下現象は、物体への液体の付着力によって発生する。この付着力は物体との境界面に対する液体の粘り気であり、境界面の近傍のみで発生する。

付着力が液体の分子間力により発生すると捉えれば、付着力の大きさは物体との接触面の大きさに依存する。液体と物体の接触面の大きさは、液体と物体の界面張力や、物質表面の微細形状などが関係している。また、付着力は温度や液体の性質、物体の性質による影響を受けるが、ここでは温度を一定とし、その他の影響をまとめて界面張力係数で表現する。

#### 3.2.2. 付着力のモデル

ここでは、本研究で用いる界面張力係数と付着力のモデル化について述べる。

提案手法では、界面張力係数によって制御された付着力を用いて、物体表面に水滴が付着する現象を再現する。界面張力係数は境界面の親水性を表す指標であり、液体と固体間の界面張力と液体と気体間の界面張力との関係や、物体表面の微細形状などに依存する。この係数によって、物体表面の物性や微細形状にもとづく水の付着力を制御する。ただし、本研究ではロータス効果などの特殊な表面形状によって発生する現象は考慮しない。また、付着力は液体や物体の種類や温度によって異なるが、本研究では液体を水に限定し、その温度も一定とした。そのため、付着力の大きさは物体表面の特性のみに依存する。

本研究では雨染みを生成する物体表面は静止しているとして、雨が物体へ付着する力のみを考える。内田らは物体を図 2 のように物体粒子の集合で表し、水の付着力  $\mathbf{f}_{i,wall}$  を式 (2) により影響半径内の物体粒子の引力の和から求めている。引力の影響半径は非常に小さいため、引力はこの範囲内で一定としている。

$$\mathbf{f}_{i,wall} = k_w \sum_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \quad (2)$$

ここで  $k_w$  は界面張力係数、 $\mathbf{x}_j$  は影響半径内にある物体粒子の位置ベクトル、 $\mathbf{x}_i$  は水の粒子の位置ベクトルである。

本研究では、このモデルを雨滴の付着力として採用した。ただし、物体境界面からの引力の算出には、物体の境界面に物体粒子を配置して式 (2) を適用する代わりに、壁重み関数を用いた。式 (5) は式 (2) を、壁重み関数を用いて表したものである。式 (5) の導出については、後の壁重み関数による壁境界条件の事前計算の説明において示す。

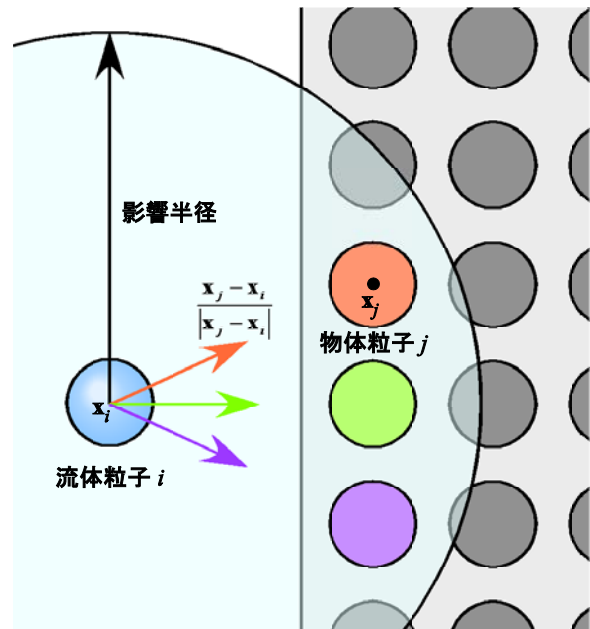


図 2 付着力と粒子

#### 3.2.3. 壁重み関数による壁境界条件の事前計算

壁重み関数による壁境界条件の事前計算では、まず Harada らの手法[4]について述べ、次に本研究による拡張として、付着力の壁重み関数によるモデル化について説明する。

提案手法では、壁重み関数によって壁境界条件を事前計算することにより、衝突の際の計算量を削減する手法を取り入れた。この前提条件として、汚れを付加する物体形状は動かないものとする。衝突判定の対象となる物体の境界面に配置した物体粒子を、以後、壁粒子と呼ぶことにする。

一般的に壁境界は壁粒子を配置して計算する[11]。しかしこの計算手法では多数の壁粒子を生成しなければならない。また壁粒子を用いて境界を表現すると、連続した壁の形状を正確に表せないという問題点もある。そこで Harada らは、壁が流体の粒子に与える影響を、壁までの距離に依存すると仮定し、流体の粒子が受ける壁粒子からの影響を壁までの距離と流体の粒

子に最も近い壁の法線ベクトルから求める手法を開発した[4].  
 提案手法はこの手法に倣って、雨滴の付着力を壁重み関数により算出する. 引力の影響半径は物体形状に対して非常に小さいため、壁粒子は水の粒子から壁境界に下ろした垂線に対して垂直かつ均一に並んでいるとする. このとき、壁境界までの距離が決まると、水の粒子に対する壁粒子の配置も一意に定まる. 水の粒子から壁粒子に向けた単位ベクトルの総和は壁粒子の配置で決定されるため、水の粒子から壁境界までの距離が決まれば、水の粒子に働く付着力も一意に定まる. そのため、付着力の式 (2) の水の粒子から壁粒子に向けた単位ベクトルの総和の部分の絶対値は、次式のように水の粒子から壁までの距離  $|r_{iw}|$  の関数となる. これを付着力の壁重み関数という.

$$Z_{wall}^w(r_{iw}) = \left| \sum_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \right| \quad (3)$$

また、壁粒子が水の粒子から壁境界に下した垂線に垂直かつ均一に並んでいるという条件から、水の付着力は水の粒子から壁境界に下した垂線、すなわち壁境界の法線に対して、対称の位置に存在する壁粒子同士で打ち消し合い、法線ベクトル方向の成分だけが残る. したがって、壁境界の法線ベクトルを  $\mathbf{n}$  とすると、式 (3) は次のように表すことができる.

$$Z_{wall}^w(r_{iw}) = \sum_j \frac{-\mathbf{x}_j \cdot \mathbf{n} + \mathbf{x}_i \cdot \mathbf{n}}{|\mathbf{x}_j - \mathbf{x}_i|} \quad (4)$$

これを用いて式 (2) を書き換えて、次式を得る.

$$\mathbf{f}_{i,wall} = -k_w \mathbf{n} Z_{wall}^w(r_{iw}) \quad (5)$$

衝突判定は水の粒子が壁粒子から受ける影響にもとづいておこなうため、任意の粒子において、その壁までの最短距離と壁境界の法線ベクトルを求める必要がある. 提案手法では、これに距離関数を用いた. 距離関数とは、計算領域に境界が定義された時に、その境界までの最短距離を表す関数である.

提案手法ではモデルの変形を考慮しないため、距離関数を事前にテーブル化しておくことで、衝突判定時の計算量を減らすことができる. また壁境界の法線ベクトルも、距離関数のテーブルを作成する際に同時に算出される. 衝突判定はこの前計算で得た結果を参照することにより、シミュレーション時の計算時間を短縮することができる.

### 3.3. 汚れのモデル化

#### 3.3.1. 汚れの分類

本研究では、非水溶性の汚れと、水溶性の汚れの二つの観点から汚れをモデル化する (図 3).

非水溶性の汚れの生成は以下の手順で行う. まず、対象物を構成する面の法線の向きにより、変化を付けた非水溶性の汚れを付着させる. 付着した汚れを水の粒子により洗い流す. 洗い流された汚れは、運搬され、別の場所に堆積する.

水溶性の汚れの生成手順は次のようになる. 水溶性の汚れは初めから水に含まれているものとする. 水と対象物が接している部分に、水溶性の汚れを付着させる.

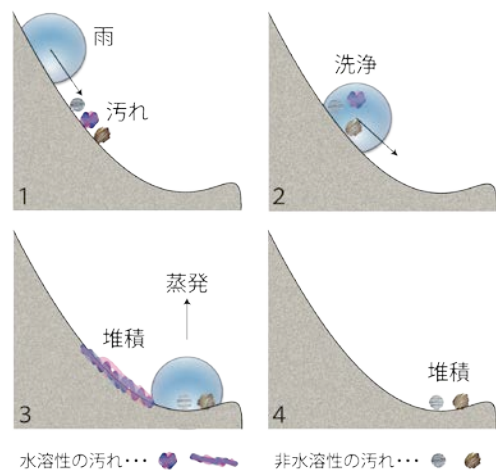


図 3 雨と汚れの相互作用

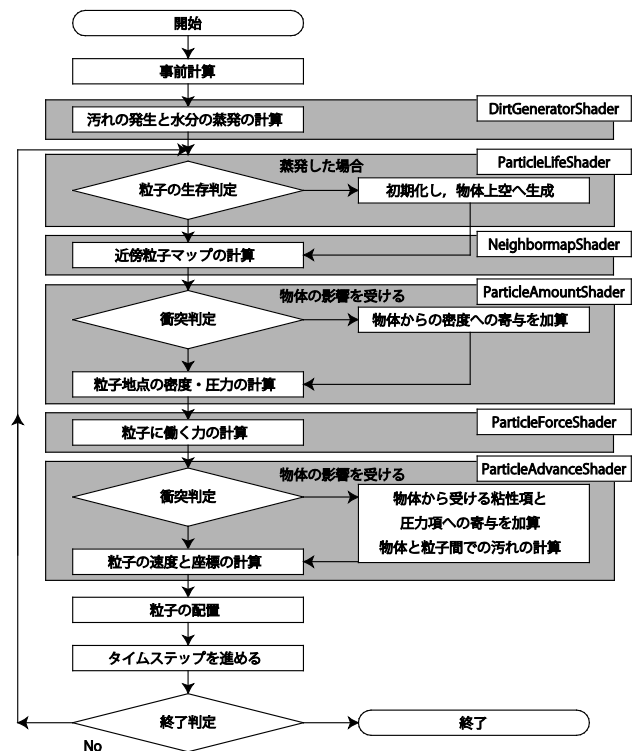


図 4 シェーダの構成とシミュレーション手順

#### 3.3.2. 汚れの移流のモデル

本研究では汚れの移流のモデルに、Dorsey らのモデル[1]を用いる. これは水彩絵の具に用いられる手法を応用した手法であり、水と物質表面の汚れと水分の移流を考慮することにより、水に含まれる汚れの濃度の違いを表現できる. このモデルにより、図 3 に示すような洗浄や堆積の作用を表現できる.

#### 3.4. 近傍粒子探索

SPH では、粒子間の相互作用を計算するため、近傍粒子探索を行う必要がある. 提案手法では、三次元格子を用いた近傍粒子探索法[3]を採用した.

### 4. GPUによる実装

本章では、提案手法の GPU 上への実装方法について述べる.

提案手法では、一般的な流体シミュレーションに対して、「汚れ」に関する物理量を取り扱うために、データ構造やシミュレーション手法もこれに対応した拡張を行っている。

シミュレーションを実行するシェーダの構成と手順を図 4 に示す。事前計算では、水の粒子の生成や距離テクスチャ、壁重み関数テクスチャ、汚れの発生源テクスチャの生成を行う。この手順では、シェーダの計算結果を Feedback Buffer (FB) に出力する GPU の機能である Transform Feedback Buffer (TFB) を利用する。FB は TFB の出力先として用いる GPU 上のメモリであり、これに GPU への入力データを保持する Vertex Buffer Object (VBO) を割り当てる。また Texture Buffer Object (TBO) は、シェーダからテクスチャのようにランダムにデータを参照することができる GPU 上のメモリである。TBO は複数のシェーダ間から参照できる共有メモリとして使用できる。

#### 4.1. 粒子のデータ構造

水の粒子が保持する物理量は、粒子の座標、粒子の速度、粒子の密度、粒子の圧力、粒子に働く力、水分の量、汚れの量の 7 個である。複数の汚れの要因に対してシミュレーションを行う場合は、粒子の物理量をその数だけ用意する必要がある。他粒子との干渉の計算に必要な物理量は、座標、速度、密度、圧力の四つである。他の三つの粒子に働く力、水分の量、汚れの量は他粒子との干渉の計算には用いない。

以下に水の粒子のデータの格納方法と計算手順について説明する。他粒子との干渉の計算に必要な物理量のデータは、VBO と TBO に格納する。TBO に格納したデータは、パーテックスシェーダからテクスチャを参照する機能である Vertex Texture Fetch (VTF) により参照できる。

計算の手順は次のとおりである。まず、頂点番号を頂点属性データとしてパーテックスシェーダに入力する。パーテックスシェーダでは、入力された頂点番号もとに TBO を参照し、粒子のデータを取得して計算を実行する (図 5)。得られた結果は TFB により FB に格納した後、FB に割り当てた VBO の内容を TBO にコピーする。次のステップでは、この TBO から粒子の物理量のデータを取得する。

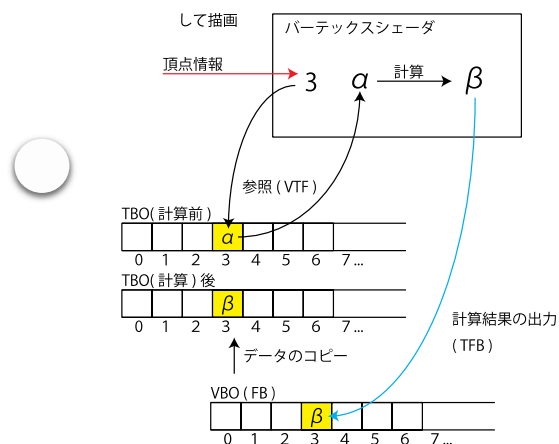


図 5 他の粒子を参照するデータの計算手順

計算の際に他の粒子の物理量を参照する必要がないデータに

ついては、二つの VBO のダブルバッファリングを用いる (図 6)。これは、まず一方の VBO からデータを入力してシェーダで処理を行う。その結果を TFB によりもう一方の VBO に格納する。その後、この二つの VBO の役割を入れ替える。すなわち、入力に使った VBO を出力先の FB として使い、出力先の FB として使用した VBO を GPU の入力に用いる。

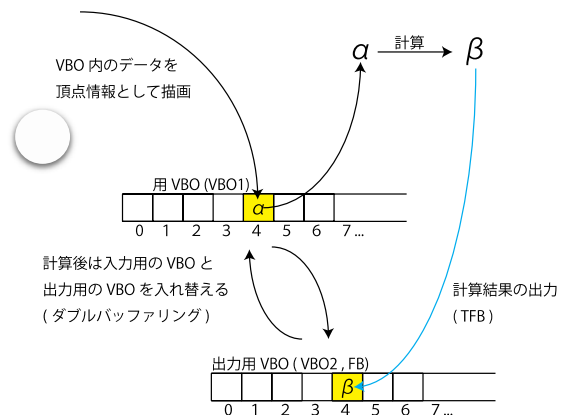


図 6 他の粒子を参照しないデータの計算手順

#### 4.2. 雨滴の発生と蒸発

雨滴の発生と蒸発のサイクルを再現するために、水の粒子の生存期間を管理するステップを用意する。このステップでは、それぞれの粒子が持つ水分の量が一定の値以下になった場合に、粒子が保持する物理量のデータを初期化する。この処理により、蒸発により水分を失う水の挙動を再現する。

初期化する粒子の物理量は、水分の量、粒子の座標、粒子の速度、粒子の持つ汚れの量の四つである。残りの三つの物理量はタイムステップ毎に値を計算し直している。

初期化された粒子のデータは、新たに降る雨滴としてシーンの上空に配置する。この粒子が保持する物理量のデータには初期値を設定する。粒子を生成する位置は、あらかじめ XorShift 法[13]によって生成した乱数をテクスチャに格納し、そのテクスチャを参照して決定している。

#### 4.3. 流体シミュレーション

流体シミュレーションは原田らの手法[5]を応用して、GPU 上に実装した。原田らの手法では粒子の物理量をテクスチャメモリに格納するが、提案手法では VBO を用いた。そして、パーテックスシェーダで計算した物理量を TFB により VBO に格納するが、その際にラスタライザを起動させずに次のステップに移ることで、GPU を流体シミュレーションの計算に専念させることができる。

また、原田らの手法では近傍粒子マップの生成の際に深度テストとステンシルテストを行うが、実験に用いた環境ではオフスクリーンのフレームバッファである Frame Buffer Object (FBO) に深度バッファとステンシルバッファを同時に追加できなかつたため、深度バッファを分割して、ステンシルバッファと共有するという手法[14]を用いた。

#### 4.4. 衝突判定

衝突判定の実装について述べる。原田らの手法[4]に付着力のモデルを組み合わせ、GPU 上に実装した。

粒子と物体の衝突判定のために、事前計算で距離関数と計算領域のある点における最も近い壁の法線ベクトルを算出し、距離テクスチャとしてテクスチャメモリに格納する。また壁重み関数は、あらかじめ壁からの距離に対応した値を求めて、一次元の壁重み関数テクスチャとしてテクスチャメモリに格納する。これにより衝突判定において、距離関数と壁の法線ベクトルを距離テクスチャから参照し、その距離に応じた壁重み関数を壁重み関数テクスチャから参照するだけで壁の影響を計算できる。

この具体的な手順は、粒子位置の更新処理において、前のタイムステップの粒子の座標とその位置の距離テクスチャから、壁境界までの距離を取得し、それにもとづいて衝突判定を行う。このとき粒子が壁に衝突していれば、壁重み関数テクスチャを参照して応力を計算する。提案手法では、ここで圧力項、粘性項、付着力の壁からの影響を加えている。その結果、壁からの影響を受ける位置にある粒子については衝突判定を省略する。

#### 4.5. 汚れの計算

物体表面に付着する汚れと水分量は、二次元テクスチャの集まりであるスライステクスチャに格納する。この汚れを三次元テクスチャとして物体表面にマッピングして結果を得る。

物体表面上の汚れの発生量と水分の蒸発量は、汚れの発生率と水分の蒸発率を格納した汚れの発生源テクスチャを参照して、タイムステップごとに求める。水の粒子の汚れのデータは、その発生時に粒子に与える。これらのデータを参照して GPU 上で衝突を検出した時に、粒子と物体の汚れの移流を計算する。

汚れの移流のモデルは Dorsey らの手法[1]を用いる。このモデルは物体と粒子が接触した場合の処理と、全ての物体表面に対する処理が混在するため、一つのシェーダで計算することはできないため、この計算の手順を次のように分割して実装した。

- 衝突の際に計算される粒子と物体間の水分と汚れの移流の部分は、粒子位置を更新する処理に実装する。
- 全ての物体表面に対して計算する物体上の水分の蒸発と汚れの発生の部分は、雨染みを生成する処理に実装する。

#### 5. 実験結果

提案手法により Stanford Bunny のモデル上に雨染みを生成した結果を図 7 に示す。各テクスチャの解像度は  $128^3$  とし、右の図ほど界面張力係数が高い。また、付着力の影響半径は流体粒子同士の影響半径と等しく、各軸の範囲が  $[-1,1]$  の空間において格子間隔より若干小さい  $0.01$  とした。

界面張力係数が低いと雨が物体表面によくなじむため、雨が直接降り注ぐ部分は洗浄されるが、雨水の流量の少ない部分への汚れの定着が多くなる。一方、界面張力係数が高くなると、雨水の流量の少ない部分への汚れの定着は少なくなっており、洗浄された結果になる。また、雨が流れる部分の汚れが洗い流されていることから、非水溶性の汚れの水が流れるところに発生しにくいという特徴が表れている。

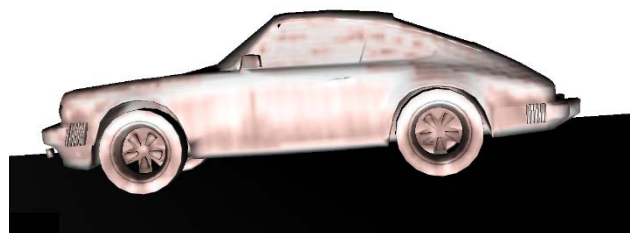


図 8 形状の詳細が雨染みに反映されない例

表 1 GPU 実装と CPU 実装の処理時間

	1024 particles	8196 particles
CPU	22.1 ms	831.3 ms
GPU	7.2 ms	23.8 ms
CPU/GPU	3.1 倍	34.9 倍

図 8 は同じ条件で車のモデルに雨染みを生成した結果を示す。この結果では、ドアミラー付近に期待される縦方向の細かい雨染みが明確には生成されなかった。これは高速化のための空間分割や距離関数、壁重み関数などの解像度が不足していることが原因だと思われる。

表 1 に GPU による実装と CPU による実装の処理時間を示す。それぞれで 1024 粒子と 8196 粒子を用いたシミュレーションを行った。1024 粒子の場合は CPU 上での実装に対して GPU 上での実装は 3.1 倍高速化された。また 8196 粒子の場合は 34.9 倍高速化された。

#### 6. おわりに

本稿では SPH 法に付着力を導入し、それを界面張力係数により制御することで、雨滴の付着力の違いを反映した雨染みの模様を生成する手法を提案した。この手法では、雨滴の付着力の算出に壁重み関数を用いて一部の処理を事前計算することにより、シミュレーション時の処理時間を短縮した。また処理を

GPU 上に実装することにより、CPU 上での実装に比べて処理速度を改善した。実験結果より、提案手法を用いて雨滴の付着力に応じて異なる汚れを生成できることが確認できた。また、GPU 上への実装により、CPU 上への実装に比べて処理速度を大幅に向上できることを確認した。これは従来手法[11]よりも総粒子数と斜面上での流体の流れにおいて有効である。

しかし、提案手法の現在の実装では、壁重み関数で用いる距離関数のテーブル化に数時間を要する。このため、提案手法はシーン中の物体の移動や変形に対応できない。これに関しては、Crassin らによるレンダリング手法[16]で用いられている Octree を生成するボクセル化アルゴリズムを導入し、Octree を用いて距離関数を動的に算出する手法の導入を検討している。

また提案手法により生成された雨染みの模様は、物体の形状を十分に反映しているとは言えない。この原因は、提案手法が GPU 上に多量のメモリを必要とするために、テクスチャの解像度を十分に高められなかったことにある。このため、流体と物体表面の接触部分の精度を十分に高めることができない。ただし、これも Octree を用いて解像度を適応的に制御することにより改善できると考えている。

このほか、提案手法の現在の実装では、雨は重力方向に不規則かつ均一に降り注ぐものとしているが、この雨の方向と分布は任意に変更可能である。これを変更することにより風の影響を表現することも不可能ではないが、それを雨染みの形状に反映するためには、雨滴が物体表面に付着後の雨滴の外力に風の力も加える必要がある。しかし、雨染みの形成には一定の時間がかかることを考えると、それに対する風の力の影響は重力ほど支配的ではないと考える。

本研究では、汚れを水溶性のものと非水溶性のものに分類した。これをもとに、提案手法では非水溶性の汚れをあらかじめ物体表面に付着しておき、それを水溶性の汚れを含んだ水で洗い流すという実装を行った。この非水溶性の汚れの初期分布は物体表面に対して均一にしているが、この初期状態が最終的な結果に与える影響を検証すべきである。

物体表面上の水の挙動の再現の点では、メッシュのパラメータ化により、これを二次元の流体シミュレーションを行う手法[15]が優れている。この手法は水滴が滴るように水が一旦物体表面から離れる状況を再現できないが、二次元のシミュレーションのため計算速度も速く、必要となるデータ領域も少ない。また、流体と物体表面の接触を精度よく計算する必要がないため、より物体形状を反映した結果が得られると思われる。また、提案手法の付着力が汚れの移流の度合いにも影響を与えてしまうという問題点も、この手法により解決できると考えられる。

最後に今後の課題について述べる。提案手法は界面張力係数による付着力の違いに応じて、異なる雨染みの形状を生成できる。界面張力係数は物体表面上で必ずしも均一とはならないが、提案手法は場所により付着力の異なる物体に対するシミュレーションも可能である。また、物体表面では浸食や傷等の経年変化や、温度によっても付着力は変化する。付着力を決定する界面張力係数を動的に計算することで、さらに複雑な汚れの再現が可能になる。

## 参考文献

- [1] J. Dorsey, H. K. Pedersen, P. Hanrahan, Flow and Change in Appearance, SIGGRAPH 96 Conference Proceedings, pp. 411-419, 1996
- [2] Y. Chen, L. Xia, T. T. Wong, X. Tong, H. Bao, B. Guo, H. Y. Shum, Visual Simulation of Weathering By g-ton Tracing, ACM Transactions on Graphics, Vol. 24, No. 3, pp. 1127-1133, 2005
- [3] 越塚, 粒子法シミュレーション 物理ベースCG入門, pp. 48-49, 培風館, 2008
- [4] T. Harada, S. Koshizuka, Y. Kawaguchi, Smoothed particle hydrodynamics in complex shapes, Proceedings of Spring Conference on Computer Graphics, pp. 235-251, 2007
- [5] T. Harada, S. Koshizuka, Y. Kawaguchi, GPUを用いた Smoothed Particle Hydrodynamics の高速化, Visual Computing / グラフィクスとCAD合同シンポジウム, 175-180, 2007
- [6] 内田, 岩崎, 融解後の水を考慮した粒子ベース氷塊融解シミュレーション, Visual Computing / グラフィクスとCAD合同シンポジウム 2010, No. 5, 2010
- [7] 三浦, 外壁汚れ防止技術の動向, 建材試験情報, Vol. 45, No. 2, pp. 6-13, 2009
- [8] 橋高, 建築物外壁仕上材料の汚染の評価方法に関する研究-3-外壁仕上材料の汚染の促進試験方法, 日本建築学会構造系論文報告集, pp. 15-24, 1989
- [9] J. J. Monaghan, Smoothed particle hydrodynamics, Annu Rev Astrophys, 543-574, 1992
- [10] M. Müller, D. Charypar, M. Gross, Particle-based fluid simulation for interactive applications, Proc. of Siggraph Symposium on Computer Animation, pp. 154-159, 2003
- [11] J. P. Morris, P. J. Fox, Y. Zhu, Modeling low reynolds number incompressible flows using sph, Journal of Computational Physics 136, 214-226., 1997
- [12] 田中, 床井, 付着力を考慮した雨による汚れのビジュアルシミュレーション, NICOGRAPH 2012 論文集, 芸術科学会, pp. 15-24, 2012
- [13] G. Marsaglia, Xorshift RNGs, Journal of Statistical Software, Vol. 8, No. 14, pp. 1-9, 2003
- [14] BlitFrameBufferExt and depth + stencil, General Purpose Computation on GPUs, <http://www.gpgpu.org/forums/viewtopic.php?p=18192>
- [15] 師, 岩崎, 土橋, 西田, 表面上の流れのためのパーティクルベースリアルタイム流体シミュレーション, 映像情報メディア学会誌, Vol. 61, No. 10, pp. 1457-1462, 2007
- [16] C. Crassin, F. Neyret, M. Sainz, S. Green, E. Eisemann, Interactive Indirect Illumination Using Voxel Cone Tracing, Computer Graphics Forum, Vol. 30, Issue 7, pp. 1921-1930, 2011

### 田中 将貴



2011年和歌山大学システム工学部デザイン情報学科卒業。2013  
年和歌山大学大学院システム工学研究科博士前期課程修了。同  
年 iCAD 株式会社入社。修士(システム工学)。コンピュータグ  
ラフィックスと物理シミュレーションに興味を持つ。

### 床井 浩平



1984年豊橋技術科学大学工学部情報工学課程卒業。1986年豊橋  
技術科学大学大学院工学研究科情報工学専攻修了。同年和歌山  
大学経済学部助手。1997年同大学システム工学部助教授。現在  
准教授。2002年博士(工学, 大阪大学)。コンピュータグラフィ  
ックスの研究に従事。電子情報通信学会, 情報処理学会, 映像  
情報メディア学会, 芸術科学会, ACM会員。