

# 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

Real-time Particle-based Method for Simulating Fluid Flows on Surfaces

師 芳卓<sup>†</sup>, 岩崎 慶<sup>††</sup>, 正会員 土橋 宜典<sup>†††</sup>, 西田 友是<sup>†</sup>

Yoshitaka Moro<sup>†</sup>, Kei Iwasaki<sup>††</sup>, Yoshinori Dobashi<sup>†††</sup> and Tomoyuki Nishita<sup>†</sup>

**Abstract** Physically based fluid simulation is one of the most important research topics in computer graphics. Although many methods have been developed that use 3D or 2D grids to simulate fluids, there are few methods for simulating fluid flows on surfaces. We present a real-time particle-based simulation method for the flows on surfaces. The surfaces are parameterized in a 2D parameter domain and physical quantities of particles are calculated in the parameter domain. The flows are simulated by updating the physical quantities of the particles, and the simulation is accelerated with a GPU. Our method simulates the fluid flows of 30,000 particles in real-time.

キーワード：流体シミュレーション, パーティクルベース, GPU, 表面上の流れ

## 1. ま え が き

水や煙, 炎などに代表される流体の動きをシミュレーションする研究は, コンピュータグラフィックスの分野において重要な研究の一つであり, 今までにさまざまな手法が提案されてきた<sup>2)-4)</sup>. しかしながらそれらの手法では3次元グリッド領域内での流体シミュレーション方法であり, 表面上の流体をシミュレーションする方法は少ない.

表面上の流体のシミュレーションは, 地球上の大気の流れなど実世界における自然現象だけでなく, 複雑な模様のテクスチャを生成する手法としても利用することができ, 応用分野が広い研究である. 表面上の流れをシミュレーションした方法としては, 以下のような研究がなされてきた. Stamは, 任意のトポロジーからなる物体表面をCatmull-Clark四辺形パッチで表現し, 表面上の流体をシミュレーションする手法を提案した<sup>1)</sup>. Shiらは, 非粘性かつ非圧縮な流体の表面上の流れをシミュレーションする手法を提案した<sup>11)</sup>.

Fanらは, Unstructured Lattice Boltzman Modelを用いて, 表面上の流れをシミュレーションした<sup>12)</sup>. Liuらは, 水の浸透を考慮した表面上の流体のリアルタイムシミュレーション法を提案した<sup>13)</sup>. これらの手法は, 効率的に表面上の流れをシミュレーションすることができるが, グリッドベースのシミュレーション方法であるため, グリッドベースシミュレーションの欠点である界面の大変形の扱いが難しく, 移流項に起因する数値拡散を生じる, という問題点がある.

一方, 移流項に起因する数値拡散が生じない, 界面の変形を扱いやすいなどの利点により, SPH(Smoothed Particle Hydrodynamics)法やMPS(Moving Particle Semi-implicit)法といったパーティクルベースの解法が近年注目されている. Mullerらは, SPH法を用いて流体の動きをインタラクティブにシミュレーションする手法を提案した<sup>5)</sup>. Premozeらは, MPS法を用いて非圧縮流体をシミュレーションする手法を提案した<sup>6)</sup>. Mullerらは, 異なる流体同士の相互作用をSPH法でシミュレーションする手法を提案した<sup>7)</sup>. Clavetらは, 粘性の高い流体のためのパーティクルベースシミュレーション法を提案した<sup>8)</sup>. 天田らは, Mullerらの手法を拡張し, GPUを利用してSPH法を高速に計算する手法を提案した<sup>9)</sup>. Kolbらは, GPU上でパーティクルベースシミュレーションを計算する手法を提案した<sup>10)</sup>.

しかしながら, これらの手法は, 主に3次元空間上の流体を扱っており, 表面上の流れのためのパーティクルベースのシミュレーション方法は提案されていなかった. その

2007年3月1日受付, 2007年6月4日再受付, 2007年6月27日採録

<sup>†</sup> 東京大学 大学院 情報理工学系研究科 コンピュータ科学専攻

(〒113-0033 文京区本郷 7-3-1, TEL 03-5841-4106)

<sup>††</sup> 和歌山大学 システム工学部

(〒640-8510 和歌山市栄谷 930, TEL 073-457-8083)

<sup>†††</sup> 北海道大学 大学院 情報科学研究科 メディアネットワーク専攻

(〒060-0814 札幌市北区北 14 条西 9 丁目, TEL 011-706-6530)

<sup>†</sup> Faculty of Engineering, The University of Tokyo

(7-3-1, Hongo, Bunkyo-ku, Tokyo 105-0011, Japan)

<sup>††</sup> Faculty of Systems and Engineering, Wakayama University

(930, Sakaedani, Wakayama, 640-8510, Japan)

<sup>†††</sup> Graduate School of Information Science and Technology

(N-14 W-9 Kita-ku, Sapporo, 060-0814, Japan)

で本論文では、表面上の流体をパーティクルを用いてシミュレーションする方法を提案する。

本論文では、物体表面をパラメータ表現し、パラメータ空間で SPH 法を計算する手法を提案する。本論文で扱う流体は、3次元空間ではなく物体の表面上を流れるため、2次元シミュレーションと同じ計算量でシミュレーションを行うことが可能である。さらに、パラメータ空間での SPH 法の計算を GPU(Graphics Processing Unit) を用いて、高速に行う方法を提案する。本研究によって、31,000 粒子で表現される流体の表面上の流れを、63fps(frames per second) の速度でシミュレーションすることが可能となる。

本論文の構成は以下のとおりである。2章において表面上の SPH 法の計算手法について説明する。3章において GPU を用いたパラメータ空間でのシミュレーション方法について述べる。4章で本研究の結果画像を示し、最後に5章でまとめと今後の課題について述べる。

## 2. 表面上における SPH 法

本研究は、Muller らの提案した SPH 法による流体シミュレーション法<sup>5)</sup>を拡張し、表面上の流れを計算する。そのため、まず SPH 法の概要について述べ、その後表面上の流体をシミュレーションするための SPH 法について説明する。

### 2.1 SPH 法の概要

パーティクル  $i$  の速度  $\mathbf{v}_i$  は以下のナビエ-ストークス方程式を解くことによって計算される。

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

ここで、 $\rho_i$  は粒子  $i$  の密度、 $p_i$  は圧力、 $\mu$  は粘性係数、 $\mathbf{F}_i^{\text{ext}}$  は外力である。パーティクルベースシミュレーションでは、パーティクルが流体とともに動くので、移流項は不要である。式(1)の第1項を圧力項  $\mathbf{F}_i^p = -\nabla p_i$ 、第2項を粘性項  $\mathbf{F}_i^v = \mu \nabla^2 \mathbf{v}_i$  と呼ぶ。式(1)を用いて、各タイムステップで加速度  $\mathbf{a}_i$  を計算し、加速度から速度  $\mathbf{v}_i$  および位置  $\mathbf{r}_i$  を更新することにより、流体の振る舞いをシミュレーションする。

SPH 法による定式化では、任意の位置  $\mathbf{r}$  での圧力や速度といった物理量  $A(\mathbf{r})$ 、 $A$  の勾配およびラプラシアンは、近傍パーティクル  $j$ (位置  $\mathbf{r}_j$ ) との距離  $r = \|\mathbf{r} - \mathbf{r}_j\|$  によるスムージングカーネル関数  $W(r, h)$  を用いて以下のように表される。

$$A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(r, h) \quad (2)$$

$$\nabla A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(r, h) \quad (3)$$

$$\nabla^2 A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(r, h) \quad (4)$$

ここで、 $m_j$  はパーティクル  $j$  の質量、 $A_j$  はパーティクル

$j$  の物理量、 $\rho_j$  はパーティクル  $j$  の密度、 $h$  はカーネルの有効半径である。密度、圧力項、粘性項の計算にはそれぞれ異なるカーネル関数  $W_{\text{poly6}}$ 、 $W_{\text{spiky}}$ 、 $W_{\text{vis}}$  を用いる<sup>5)</sup>(付録参照)。

#### (1) 密度

パーティクル  $i$  の密度  $\rho_i$  は式(2)から以下の式で求められる。

$$\rho_i = \sum_j m_j W_{\text{poly6}}(\|\mathbf{r}_i - \mathbf{r}_j\|, h) \quad (5)$$

#### (2) 圧力項

圧力項  $\mathbf{F}_i^p$  は式(3)の物理量に  $(p_i + p_j)/2$  を代入することによって計算される。

$$\mathbf{F}_i^p = - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W_{\text{spiky}}(\|\mathbf{r}_i - \mathbf{r}_j\|, h) \quad (6)$$

ここで、パーティクル  $i$  の圧力  $p_i$  は以下の式で計算する。

$$p_i = k(\rho_i - \rho_0) \quad (7)$$

ここで、 $k$  と  $\rho_0$  はパラメータである。

#### (3) 粘性項

粘性項  $\mathbf{F}_i^v$  は、パーティクルの速度の差から以下の式で計算される。

$$\mathbf{F}_i^v = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W_{\text{vis}}(\|\mathbf{r}_i - \mathbf{r}_j\|, h) \quad (8)$$

### 2.2 表面上の流体への拡張

本研究では、表面上の流体のシミュレーションをパラメータ空間で計算する。すなわち、パラメータ空間でパーティクルの位置および速度を計算し、パラメータ空間において各粒子にかかる力を計算する(図1)。そのため、あらかじめパラメータ空間  $U \in [-1, 1]^2$  から物体表面  $S \in R^3$  への変換  $\mathbf{f}: (u, v) \rightarrow (x, y, z)$  を求めておく。この変換  $\mathbf{f}$  は、物体表面をパラメータ化することによって計算される。物体表面をパラメータ化する方法としては、さまざまな手法が提案されているが、パラメータ化による歪みが少ないことから、本論文では、種数が0の物体表面については等角パラメータ化手法<sup>14)</sup>を使用し、それ以外の物体表面については stretch minimizing 法<sup>15)</sup>を使用した。

シミュレーション空間からパラメータ空間への物理量の変換はヤコビ行列  $\mathbf{J}$  を介して行われる。

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{pmatrix} \quad (9)$$

密度  $\rho_i$ 、圧力項  $\mathbf{F}_i^p$  および粘性項  $\mathbf{F}_i^v$  は以下の式で計算される\*。

$$\rho_i = \sum_j m_j W_{\text{poly6}}(d_{ij}, h) \quad (10)$$

\* 粘性項を正確に計算するためには  $\nabla W_{\text{vis}}, \partial^2 W_{\text{vis}} / \partial x^2$  を計算する必要があるが、距離0の時に発散する関数であるため式(11)により近似した。

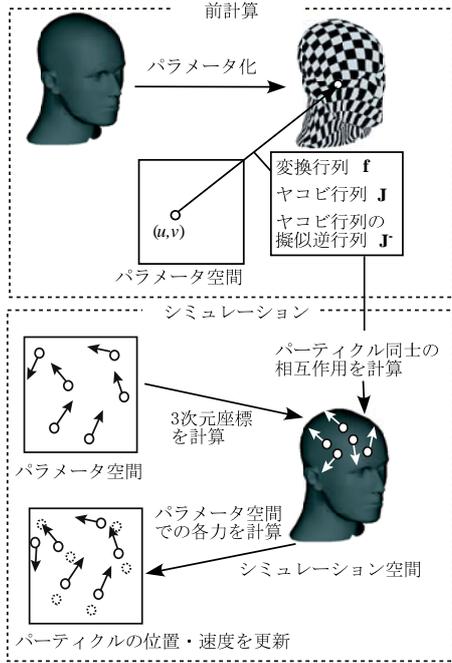


図 1 表面上の SPH 法の概要

Overview of the SPH method for simulating flows on surfaces.

$$\mathbf{F}_i^p = -\mathbf{J}^-(\mathbf{u}_i) \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W_{spiky}(d_{ij}, h) \quad (11)$$

$$\mathbf{F}_i^v = \mu \sum_j m_j \frac{\nu_j - \nu_i}{\rho_j} \nabla^2 W_{vis}(d_{ij}, h) \quad (12)$$

ここで、 $\nu_i \in R^2$  はパラメータ空間での速度、 $\mathbf{u}_i \in U$  はパラメータ空間でのパーティクル  $i$  の位置、 $d_{ij}$  はシミュレーション空間におけるパーティクル間の距離で、以下の式で計算される。

$$d_{ij} = \|\mathbf{J}(\mathbf{u}_j)(\mathbf{u}_i - \mathbf{u}_j)\| \quad (13)$$

$\mathbf{J}^-$  はヤコビ行列の擬似逆行列で、 $\mathbf{J}^- = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$  で計算される。本研究では、パラメータ空間において各項を GPU で高速に計算する。

### 3. GPU を用いた SPH の高速計算法

#### 3.1 概要

パーティクル  $i$  に働く圧力項や粘性項は、近傍のパーティクルの物理量にパーティクル間の距離  $r_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$  に応じた重み  $W(r_{ij}, h)$  を掛けて足し合わせることで計算される (図 2(a))。そのため、天田らの方法<sup>9)</sup> では各パーティクルの近傍のパーティクルのリストを CPU で計算し、そのリストに含まれる近傍パーティクルについて各項を GPU で計算していた。近傍のパーティクルは毎ステップで移動するためリストの更新をする必要がある。

本研究では、近傍のパーティクルを探索し、パーティクル間で物理量を計算する代わりに、パラメータ空間における物理量の場を計算する。パラメータ空間に仮想的なグリッドを設定し、各パーティクルから各グリッドへの物理量を

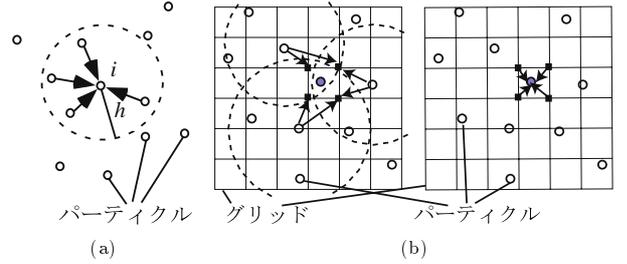


図 2 物理量の計算方法。(a) 従来法: パーティクル  $i$  の物理量は、近傍パーティクルからの寄与を計算して求める。(b) 提案手法: 仮想的なグリッドへの物理量の寄与を累積することによって、場を計算する。パーティクル  $i$  の物理量は近傍のグリッドに累積された値から補間して計算する。  
Calculation of physical quantities.

計算する (図 2(b))。すべてのパーティクルについて物理量を計算し、累積することによってパラメータ空間での物理量の場を計算する。そして、パーティクル  $i$  における物理量の計算は、 $i$  に隣接するグリッドに累積された物理量から計算する。

この計算は、GPU を利用して高速に処理される。パラメータ領域の仮想グリッドと、フレームバッファの各ピクセルを対応させる。パーティクル  $i$  からカーネル半径  $h$  以内の各グリッドへの物理量の計算は、カーネル関数  $W$  の値と物理量を掛けた値を保持した画像をフレームバッファに描画し輝度を累積することによって、GPU で処理される。パラメータ領域における物理量を計算し、グリッドでの物理量の値を保持したテクスチャを作成する。各パーティクルに作用する圧力、粘性、表面張力はそのパーティクルの近傍のグリッドでの物理量から補間して計算する。この補間の計算は、物理量を保持したテクスチャのマッピングにより GPU を用いて高速に処理される。

シミュレーションの流れは以下ようになる。

- (1) 密度場および密度の計算
- (2) 圧力場および圧力項の計算
- (3) 粘性場および粘性項の計算
- (4) 各パーティクルの加速度を計算し、速度と位置を更新

以下、各ステップについて説明する。

#### 3.2 密度場および密度の計算

各パーティクルの密度を計算するために、まずパラメータ空間における密度の場を計算する。パラメータ空間における密度場は、各パーティクルの質量  $m_i$  にカーネル関数  $W_{poly6}$  を乗じてグリッドに累積することで求められる。

カーネル関数  $W_{poly6}$  の値を計算するためには、パーティクルの位置とパラメータ領域での各グリッドとの、シミュレーション領域での距離を計算する必要がある。シミュレーション領域での距離は、各パーティクルと各グリッドの物体表面上の位置を計算することによって求められる。パラメータ空間から物体表面上の位置は変換  $f$  をあらかず行列を乗じることによって求められるが、計算コストが高い。そ

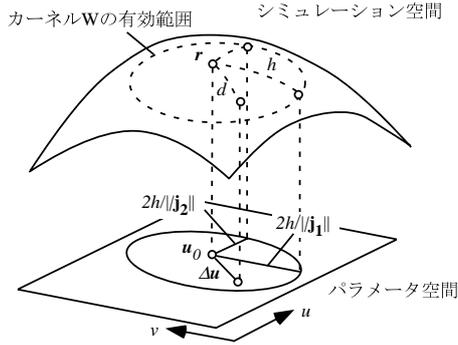


図 3 パラメータ空間におけるカーネル関数  
Kernel function in a parameter domain.

ここで本研究ではシミュレーション空間における距離を計算せずにカーネル関数の値を求める．

図 3 に示すように， $\mathbf{u}_0 = (u_0, v_0) \in U$  をカーネルの中心とし， $\Delta \mathbf{u} = (\Delta u, \Delta v)$  をカーネルの中心と計算点との差のベクトルとする．計算点とカーネルの中心の，シミュレーション空間における距離  $d$  は  $d = \|J(\mathbf{u}_0)\Delta \mathbf{u}\|$  で計算される．ここで，変換  $f$  は等角写像であるため，ヤコビ行列を  $J(\mathbf{u}_0) = (\mathbf{j}_1, \mathbf{j}_2)$  とすると， $\mathbf{j}_1 \cdot \mathbf{j}_2 = 0$  が成り立つ．そのため，距離  $d$  は以下のように計算される．

$$d^2 = \|\mathbf{j}_1\|^2 (\Delta u)^2 + \|\mathbf{j}_2\|^2 (\Delta v)^2 \quad (14)$$

この式は，パラメータ空間におけるカーネル関数の形状は， $2h/\|\mathbf{j}_1\|, 2h/\|\mathbf{j}_2\|$  を軸とする楕円になることを示している． $\mathbf{u}$  における楕円の大きさは， $\|\mathbf{j}_1\|$  と  $\|\mathbf{j}_2\|$  を計算することによって求められる．そこで，あらかじめパラメータ空間の各グリッドにおいて  $\|\mathbf{j}_1\|, \|\mathbf{j}_2\|$  の値を計算しテクスチャとして保存しておく．計算点とカーネルの中心との距離  $d$  はこのテクスチャを参照することにより求められるため，計算点におけるカーネルの値は高速に計算される．

パラメータ空間における密度場を密度場マップとして保存し，各パーティクルの密度は密度場マップから補間して計算する．各パーティクルの密度は，テクスチャとして保存し，各タイムステップで更新する．

### 3.3 圧力場および圧力項の計算

パーティクル  $i$  の圧力項は式 (11) で計算される．式 (11) はパーティクル  $i$  の圧力  $p_i$  を含んでいるため，以下のように  $p_i$  に依存する項と依存しない項に分けてグリッドに累積する．

$$\mathbf{F}_i^p = -\mathbf{J}^-(\mathbf{u}_i) \left( \sum_j \mathbf{I}_j^{p0}(\mathbf{u}_i) - p_i \sum_j \mathbf{I}_j^{p1}(\mathbf{u}_i) \right) \quad (15)$$

$$\mathbf{I}_j^{p0}(\mathbf{u}) = m_j \frac{p_j}{2\rho_j} \nabla^2 W_{spiky}(\|J(\mathbf{u}_j)(\mathbf{u}_j - \mathbf{u})\|, h) \quad (16)$$

$$\mathbf{I}_j^{p1}(\mathbf{u}) = m_j \frac{1}{2\rho_j} \nabla W_{spiky}(\|J(\mathbf{u}_j)(\mathbf{u}_j - \mathbf{u})\|, h) \quad (17)$$

パラメータ空間における圧力場は  $\mathbf{I}^{p0}$  と  $\mathbf{I}^{p1}$  を二つのテクスチャ(圧力場マップ)として保存する．パーティクル  $i$  の圧力項  $\mathbf{F}_i^p$  は，二つの圧力場マップから補間した値  $\mathbf{I}^{p0}(\mathbf{u}_i)$ ,

$\mathbf{I}^{p1}(\mathbf{u}_i)$  および圧力  $p_i$  を用いて式 (15) から計算される．

### 3.4 粘性場および粘性項の計算

粘性項も，パーティクル  $i$  の速度  $\nu_i$  に依存する項と依存しない項に分けて保存する．

$$\mathbf{F}_i^v = \mu \sum_j \mathbf{I}_j^{v0}(\mathbf{u}_i) - \mu \nu_i \sum_j \mathbf{I}_j^{v1}(\mathbf{u}_i) \quad (18)$$

$$\mathbf{I}_j^{v0}(\mathbf{u}) = m_j \frac{\nu_j}{\rho_j} \nabla^2 W_{vis}(\|J(\mathbf{u}_j)(\mathbf{u}_j - \mathbf{u})\|, h) \quad (19)$$

$$\mathbf{I}_j^{v1}(\mathbf{u}) = m_j \frac{1}{\rho_j} \nabla W_{vis}(\|J(\mathbf{u}_j)(\mathbf{u}_j - \mathbf{u})\|, h) \quad (20)$$

パラメータ空間における粘性場は，粘性場マップとして記憶する．粘性場マップは， $\mathbf{I}^{v0}$  を RGB 値に保存し， $\mathbf{I}^{v1}$  をアルファ値に保存する．

### 3.5 加速度・速度・位置の更新

式 (2) に基づいて各パーティクルの加速度を計算する．各パーティクルの速度  $\nu_i$  および位置  $\mathbf{u}_i$  をテクスチャとして保存しておくことにより，GPU 上でシミュレーションを行う．

## 4. 結果

本研究による表面流れの流体シミュレーション結果を示す．図の下向きに重力の働く空間上の物体の表面に流体を配置し，モデルを回転させた時に流体が重力に従って流れる様子をシミュレーションしている．図 4 は，本手法を 2次元平面に適用し”SPH”という文字の形をした障害物のある表面上の流体の流れのシミュレーション，図 5 は rabbit モデルの頭の部分に色の異なる流体を配置したシミュレーション，図 6 では igea モデルの後頭部の部分に色の異なる流体を配置したシミュレーションを可視化した結果である．図 4(a),(b) と (c),(d) および図 5(a)-(d) と (e)-(h) で物体の傾きを変更し，図 6 では粒子数を変えてシミュレーションを行った．\*

使用したパーティクルの数は図 4 が 11,275 個，図 5 が 15,670 個，図 6(a)-(d) が 31,041 個，(e)-(h) が 227,130 個である．密度場などのパラメータ空間における場を保持するテクスチャおよび各パーティクルの速度や位置を保持するためのテクスチャの解像度は図 6(e)-(h) が  $512 \times 512$ ，他は  $256 \times 256$  である．計算環境は，CPU が Intel Core 2 Extreme，メモリーが 2GB，GPU が nVidia GeForce 8800 GTX のマシンにおいて，図 4 が 71fps，図 6(a)-(d) が 63fps，図 6(e)-(h) が 14fps，図 5 が 65fps であった．本研究により，複雑な表面上の流れをリアルタイムにシミュレーションができる．

次に，igea モデルを用いて，場のテクスチャの解像度とパーティクル情報のテクスチャの解像度をそれぞれ変えた場合の計算量の比較を行った．(a) はパーティクルのテクスチャの解像度が  $256 \times 256$  で粒子数が 52,566 個，(b) は

\* 図 4,5,6 のシミュレーション結果は以下を参照せよ．

<http://nis-lab.is.s.u-tokyo.ac.jp/~ymoro/media2007/>

表 1 テクスチャの解像度を変化させた場合のシミュレーションのフレームレートの比較。なお、\*の付いた項目ではシミュレーションが安定しなかった。

Comparison of frame rates of simulations.

場のテクスチャ解像度	(a)	(b)	(c)
128 × 128	32	7.5*	1.9*
256 × 256	53	7.5	1.9*
512 × 512	27	14	1.8
1024 × 1024	13	7.0	3.2

512 × 512 で粒子数が 227,130 個，(c) は 1024 × 1024 で粒子数が 1,009,540 個という条件で実験を行い，表 1 にまとめた。表 1 から，GPU の特性により両者のテクスチャの解像度が一致した場合に計算速度が上がるのがわかる。また，テクスチャのサイズが 2 倍になると，計算時間が 4 倍になることがわかる。時間間隔も 2 倍細かく取る必要があるため，シミュレーションの計算量は，テクスチャのサイズの 3 乗に比例して増えることがわかる。

### 5. まとめと今後の課題

GPU を用いて高速に表面上の流体シミュレーションを行うパーティクルベースのシミュレーション方法を提案した。物体表面をパラメータ表現し，パラメータ空間上でパーティクルの相互作用を計算することによって，表面上の流れのシミュレーションを行った。パラメータ空間をグリッド分割し，各グリッドをテクスチャの各ピクセルに対応させることによって，パラメータ空間における物理量の場を GPU を利用して高速に計算した。本研究では，3 万個のパーティクルで表現される流体の表面上の流れをリアルタイムにシミュレーションすることができた。

今後の課題として，パラメータ化による歪みの軽減があげられる。また，地球表面上の大気の流れのシミュレーションへの適用があげられる。

### 【文 献】

- 1) J. Stam, "Flows on Surfaces of Arbitrary Topology", Proc. of SIGGRAPH 2003, pp.724-731 (2003)
- 2) J. Stam, "Stable Fluids", Proc. of SIGGRAPH 1999, pp.121-128 (1999)
- 3) N. Foster, R. Fedkiw, "Practical Animation of Liquids", Proc. of SIGGRAPH 2001, pp.23-30 (2001)
- 4) R. Fedkiw, J. Stam, H. Jensen, "Visual Simulation of Smoke", Proc. of SIGGRAPH 2001, pp.15-22 (2001)
- 5) M. Muller, D. Charypar, M. Gross, "Particle-based fluid simulation for interactive application", Proc. of Symposium on Computer Animation 2003, pp.154-159 (2003)
- 6) S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, R. Whitaker, "Particle-based simulation of fluids", Computer Graphics Forum, **22**, 3, pp.401-410 (2003)
- 7) M. Muller, B. Solenthaler, R. Keiser, M. Gross, "Particle-based Fluid-Fluid Interaction", Proc. of Symposium on Computer Animation, pp.1-7 (2005)
- 8) S. Clavet, P. Beaudoin, P. Poulin, "Particle-based Viscoelastic Fluid Simulation", Proc. of Symposium on Computer Animation 2005, pp. 219-228 (2005)
- 9) 天田崇, 井村誠孝, 安室喜弘, 眞鍋佳嗣, 千原國宏, "剛体との相互作用を伴う水の実時間アニメーション", 映像学誌, **59**, 10, pp. 1488-1493 (2005)
- 10) A. Kolb, N. Cuntz, "Dynamic Particle Coupling for GPU-based

Fluid Simulation", Proc. of 18th Symposium on Simulation Technique, pp.722-727 (2005)

- 11) L. Shi, Y. Yu, "Inviscid and Incompressible Fluid Simulation on Triangle Meshes", Computer Animation and Virtual Worlds, **15**, 3-4, pp.173-181 (2004)
- 12) Z. Fan, Y. Zao, A. Kaufman, Y. He, "Adapted Unstructured LBM for Flow Simulation on Curved Surfaces", Proc. of Symposium on Computer Animation, pp. 245-254 (2005)
- 13) Y.Q. Liu, H.B. Zhu, X.H. Liu, E.H. Wu, "Real-time simulation of physically based on-surface flow", The Visual Computer, **21**, 8-10, pp.727-734 (2005)
- 14) X. Gu, Y. Wang, T.F. Chan, P.M. Thompson, S.T. Yau, "Genus zero surface conformal mapping and its application to brain surface mapping", IEEE Trans. Med. Imaging, **23**, 8, pp.949-958 (2004)
- 15) S. Yoshizawa, A. Belyaev, H. Seidel, "A fast and simple stretch-minimizing mesh parameterization", Proc. of Shape Modeling International 2004, pp. 200-208 (2004)

### 付 録

本研究では，Muller の方法<sup>5)</sup>を基に，2次元平面上で正規化した次のようなカーネル関数(有効半径  $h$ )を用いている。密度の計算および表面張力の計算では以下のカーネル  $W_{poly6}$  を用いる。

$$W_{poly6}(r, h) = \frac{4}{\pi h^8} \begin{cases} (h^2 - r^2)^3 & 0 \leq r \leq h \\ 0 & otherwise \end{cases}$$

圧力項の計算には以下のカーネル  $W_{spiky}$  を用いる。

$$W_{spiky}(r, h) = \frac{10}{\pi h^5} \begin{cases} (h - r)^3 & 0 \leq r \leq h \\ 0 & otherwise \end{cases}$$

粘性項の計算には以下のカーネル  $W_{vis}$  を用いる。

$$W_{vis}(r, h) = \frac{10}{3\pi h^2} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 & 0 \leq r \leq h \\ 0 & otherwise \end{cases}$$

	<p>もろ 芳卓 2005 年，東京大学理学部卒業。2007 年，東京大学情報理工学系研究科修士課程修了。主としてコンピュータグラフィクスに関して，大気散乱のレンダリングおよび流体シミュレーションなどの研究に従事。</p>
	<p>いわさき 岩崎 慶 1999 年，東京大学理学部卒業。2001 年，同大学院新領域創成科学研究科博士前期課程修了。2004 年，同大学院新領域創成科学研究科博士後期課程修了。同年，和歌山大学システム工学部助手。2007 年，同講師。主としてコンピュータグラフィクスに関する研究に従事。科学博士。</p>
	<p>とばし 土橋 宜典 1992 年，広島大学工学部第二類(電気系)卒業。1994 年，同大学大学院工学部研究科システム工学専攻博士課程前期修了。1997 年，同専攻博士課程後期修了。同年，広島市立大学情報科学部助手。2000 年，北海道大学工学部研究科助教授。主としてコンピュータグラフィクスに関して，照明シミュレーション，景観予測などの研究に従事。工学博士。正会員。</p>
	<p>にしだ 西田 友是 1971 年，広島大学工学部卒業。1973 年，同大学大学院工学研究科修了。同年，マツダに入社。1979 年，福山大学工学部講師。1984 年，同助教授。1990 年，同教授。1998 年，東京大学理学部教授。1999 年，同大学大学院新領域創成科学研究科教授となり，現在に至る。コンピュータグラフィクスの研究に従事。工学博士。</p>

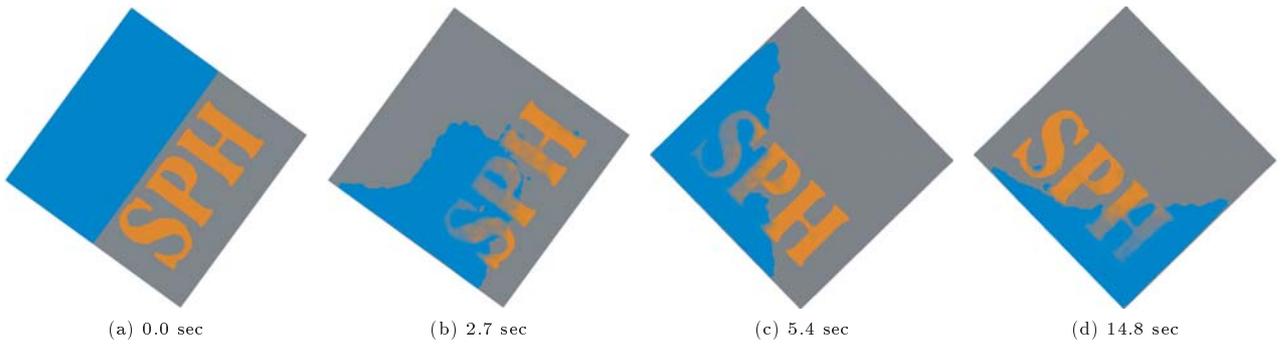


図 4 平面上に障害物を置いた場合の流れのシミュレーション結果  
Simulation results of flows on plane with obstacles.

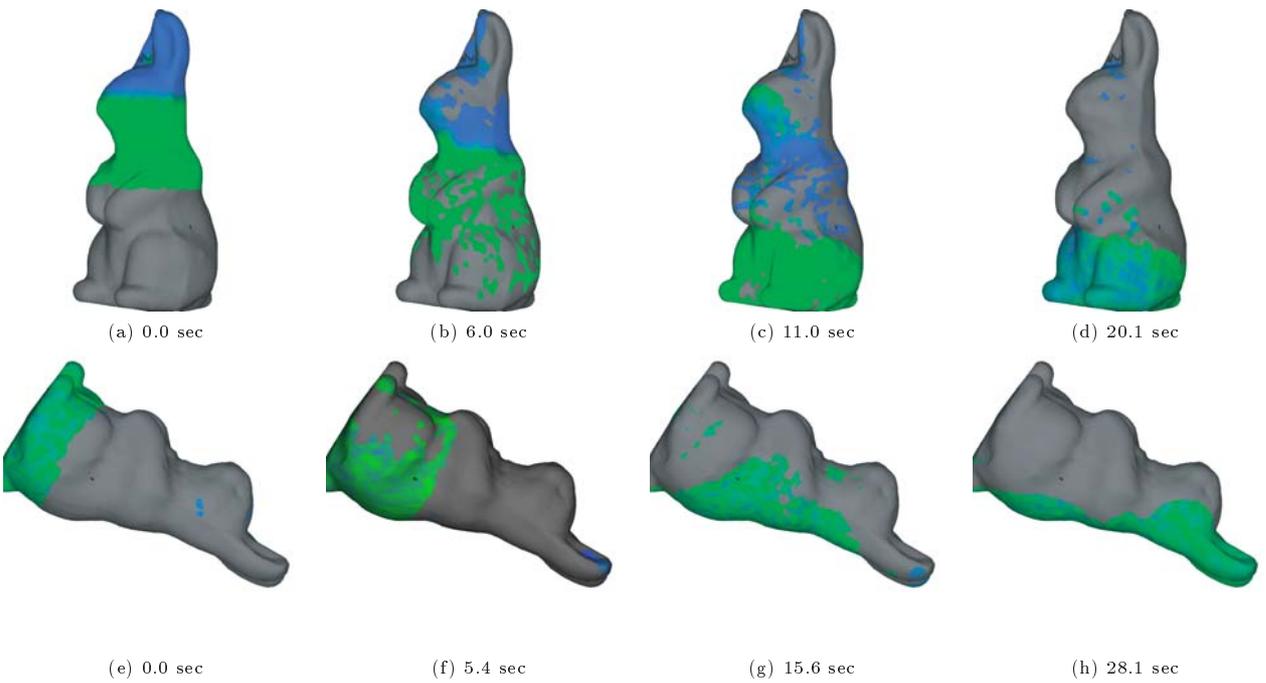


図 5 bunny モデル表面上の流れのシミュレーション結果  
Simulation results of flows on bunny model.

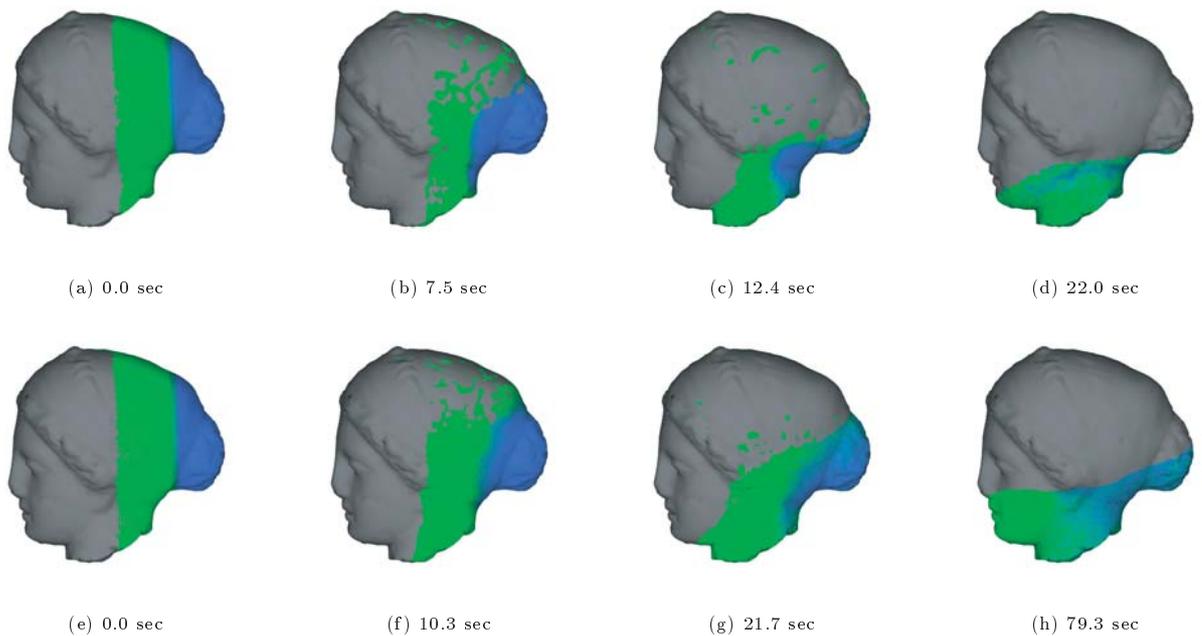


図 6 igea モデル表面上の流れの粒子数の違いによるシミュレーション結果の比較  
Comparison of simulation results of flows on igea model with different particles.