

流体と布のリアルタイム連成シミュレーション

原田 隆宏[†] 越塚 誠一[†] 河口 洋一郎[†]

本論文では粒子法の1つである Smoothed Particle Hydrodynamics (SPH) を用いた布と流体の連成をリアルタイムで計算する手法を提案する。ポリゴンで表される布と、流体粒子との相互作用を計算する際に、粒子から布までの距離を計算しなければならない。この計算では粒子からポリゴンを構成する面、辺、頂点までの距離を計算しなければならない。計算コストが高い。そこで本研究ではポリゴンの重心までの距離とポリゴンの面までの距離を用いて、流体粒子から布までの距離を近似計算する。また本論文ではこの布と流体の連成シミュレーション手法を Graphics Processing Unit (GPU) 上で全て計算を行なう手法も提案する。流体に働く布からの力と、布に働く流体からの力の計算を別々に行なうことで、GPU 上での計算を可能にした。GPU の並列性を生かすことで、数万流体粒子と数千ポリゴンからなる布との連成計算をリアルタイムで行なうことが可能になった。

Real-time Fluid Simulation Coupled with Cloth

TAKAHIRO HARADA,[†] SEIICHI KOSHIZUKA[†]
and YOICHIRO KAWAGUCHI[†]

This paper presents a real-time simulation method for coupling of cloth and fluids computed by using Smoothed Particle Hydrodynamics (SPH). To compute interaction between cloth consists of several polygons and fluid particles, the distance between cloth to the particle have to be estimated. It is computationally expensive because we have to compute the distance to the faces, edges and vertices of polygons. Therefore, we calculate an approximate distance to the cloth by using the distance to the faces and the gravitational centers of the polygons. This paper also presents techniques to perform the coupled simulation entirely on Graphics Processing Units (GPUs). Dividing the interaction computation into fluid-cloth and cloth-fluid forces makes it possible to use GPUs for simulation. By exploiting the parallelism of GPUs, we could couple simulations of several tens of thousands of fluid particles and cloth which consists of several thousands of polygons in real-time.

1. 序 論

本研究では粒子法の1つである Smoothed Particle Hydrodynamics (SPH) を用いて、リアルタイムで布と流体との連成計算を行なう手法を提案する。布を構成するポリゴンがある厚みがあると仮定して流体粒子との相互作用を計算すると、流体粒子にかかる力が空間で不連続になってしまうため、布から流体粒子にかかる力を空間で連続にする手法を提案する。また布を構成するポリゴンと流体粒子の衝突計算を高速に行うため、2個の格子を導入する。1個目の格子は流体粒子の近傍粒子探索を高速化するために用い、2個目の格子は布と流体粒子の衝突計算を行なうために用いる。本論文で述べる布と粒子法のカップリング手法は布から流体が受ける力は外力として計算されるため、

Distinct Element Method (DEM) や Moving Particle Semi-implicit (MPS) Method などの SPH 以外の粒子法にも用いることができる³⁾¹³⁾。また本手法は計算を並列処理可能であることを示し、計算の全てを Graphics Processing Units (GPUs) 上で実装する手法を述べる。これにより計算の大幅な高速化が可能になった。

2. 関連研究

ラグランジュ解法である粒子法に関しても数多く研究が行われてきた。Müller らは SPH を用いてリアルタイムシミュレーションを行ない¹⁶⁾、混相流の研究も行なった¹⁸⁾。Kipfer らは空間的に疎な粒子配置に適したデータ構造を用いて SPH を高速化して川のリアルタイムシミュレーションを行なった¹¹⁾。

流体と弾性体や薄膜との連成計算の研究も行われてきた。Génevaux らは物体を質点で表し、それらをバネで繋いだ弾性体として表現した⁵⁾。Müller らは SPH

[†] 東京大学
The University of Tokyo

を用いた流体計算と四面体メッシュで表された弾性体との連成計算を行なった¹⁷⁾。Guendelman らは薄膜とオイラー解法による流体計算のカップリングの研究を行なった⁶⁾。しかしこの手法の計算コストは高いため、リアルタイムアプリケーションへ応用するのは困難である。Chentanez らはKlingner らの手法を応用し、弾性体と流体を強連成で解いた²⁾。

また GPU の計算能力に着目した研究も多く存在する¹⁹⁾。Harada らは剛体を粒子で表現し、剛体の衝突計算を GPU を用いて高速化した^{7),21)}。GPU を用いて SPH を高速化する研究は様々な問題点があったが¹⁾¹²⁾、Harada らはこれらの問題点を解決し、GPU を用いて SPH を高速化した⁸⁾。このように SPH に対して GPU を用いて CPU と同様に計算する研究は存在せず、さらに GPU を用いて SPH と他の物体の計算をカップリングさせる研究も行なわれていない。

3. 流体シミュレーション

非圧縮流体の支配方程式は以下に示す連続の式と運動量保存の式である。

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u} + \frac{\mathbf{f}}{\rho} \quad (2)$$

であり、 $\rho, \mathbf{u}, P, \nu, \mathbf{f}$ はそれぞれ流体の密度、圧力、速度、動粘性係数そして外力である。本研究では流体を粒子で離散化し、SPH を用いて支配方程式を解いた。SPH では式 (1) を解かないため、非圧縮流体を解くことはできない。しかし圧縮性を弱めることでほぼ非圧縮な流体を解くことができる。またある座標での物理量は、その座標の近傍に存在する粒子の持つ物理量の重み付け和として求める。本研究では Müller らの計算モデルを用いて SPH の計算を行なった^{8),16)}。

4. 布シミュレーション

布と流体の相互作用から布に働く力は外力として求まる。そのため、どのような計算モデルを布に用いることもできる。本研究では布のモデルに粒子同士が2種類のバネで繋がれているモデルを用いた(図??)。そして時間積分にはオイラー陽解法よりも安定性の高いベルレ法を用いた。この手法は計算コストが低く安定性が高いため、リアルタイムシミュレーションに向いている。時刻 $t + dt$ での粒子 i の座標 \mathbf{x}_i^{t+dt} は以下のように計算される。

$$\mathbf{x}_i^{t+dt} = \mathbf{x}_i^t + d(\mathbf{x}_i^t - \mathbf{x}_i^{t-dt}) + \frac{\mathbf{F}_i dt^2}{m_i} \quad (3)$$

m_i, d, dt はそれぞれ粒子の質量と減衰係数、時間刻み幅である。 \mathbf{F}_i は外力であり、重力 $m_i \mathbf{g}$ とバネによる力 $\mathbf{F}_{i,spring}$ と流体から受ける力 $\mathbf{F}_{i,fluid}$ の和である。バネによる力 $\mathbf{F}_{i,spring}$ は以下のように計算される。

$$\mathbf{F}_{i,spring} = \sum_{j \in N_{adj}} k_{adj} (|\mathbf{x}_{ij}| - l_{adj}) \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} + \sum_{j \in N_{diag}} k_{diag} (|\mathbf{x}_{ij}| - l_{diag}) \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \quad (4)$$

k_{adj}, l_{adj} は上下左右の粒子を接続しているバネのバネ定数と自然長であり、 k_{diag}, l_{diag} は斜めの粒子を接続しているバネのバネ定数と自然長である。 \mathbf{x}_{ij} は粒子 i から見た粒子 j の相対座標である。

5. 流体と布の相互作用

5.1 空間分割

SPH 法も含めて粒子法では粒子にかかる力を計算するために近傍粒子を探索しなければならない。近傍粒子を n 個の全粒子から探索するとその計算コストは $O(n^2)$ になるため効率が悪い。そこで計算領域を3次元格子で覆い、格子のそれぞれのボクセル内に存在する粒子番号をそのボクセルに対応づけることで、近傍粒子探索を効率化することができる¹⁵⁾。本研究でも流体の近傍粒子探索においてこの手法を用いる。また布を構成するポリゴンと流体粒子の相互作用を計算するときには流体粒子とポリゴンとの最小距離を求める必要があるため、 m 個の全ポリゴンと距離を計算するとその計算コストは $O(mn)$ になってしまう。そこでこの計算も格子を導入し計算領域を分割することで効率化することができる。流体粒子はそれぞれを質点として3次元格子内の座標を計算し、そのボクセルに粒子番号を格納する。布の場合はポリゴンが大きさを持った平面であるため、複数のボクセルに面の番号を記録しなければならない。しかし本手法ではポリゴンの重心座標が存在するボクセルにポリゴンの番号を記録する。すなわち、複数のボクセルにまたがって存在するポリゴンでも、その番号は1つのボクセルにしか記録されない。このように流体と布をカップリングするために、流体粒子と布のポリゴン用の2個の格子を導入する。

本研究で用いる布を構成するポリゴンは全て同じ大きさであり、布の物理的性質により計算中に大きく面積を変えることはない。そこで布の平均の大きさに応じてボクセルの1辺の長さを調節する。流体粒子の相互作用を計算するときには1辺が粒子径の大きさのボクセルを用いて空間分割を行なうが、本研究で用いた

布ポリゴンの大きさは流体粒子の数倍の大きさであるため、布ポリゴンを格納するボクセルは1辺が粒子径の数倍のボクセルを用いた。

5.2 衝突検出

流体粒子と布との相互作用を計算するには、まず流体粒子と布との距離を計算する必要がある。流体粒子と布までの距離は粒子に最も近い位置に存在するポリゴンもしくはその辺、頂点までの距離であるため、これを探して布との距離を計算する。粒子と距離が最も小さいポリゴンを全てのポリゴンから探索する必要があるが、ポリゴンの番号が格納された格子を用いることで限られた数のポリゴンとの距離を計算するだけでよい。この処理ではまずそれぞれの粒子座標から布の3次元格子内での座標を計算する。その粒子と近距離に存在するポリゴンは、粒子が存在するボクセルとその周辺のボクセルの計27個のボクセル内にポリゴンの番号が格納されているので、それらのポリゴンとの距離を計算する。

布と流体粒子との相互作用は、布がある厚みを持っていると考え、粒子のめりこみに応じた力を加えることによって計算することができる。しかし布との距離の計算を粒子と布を構成するポリゴンとの距離を計算して最小値を取り粒子に働く力を計算すると、その力が空間で不連続になるため、粒子が突然飛び跳ねてしまうことがある。例えばポリゴンの接続部分近辺に存在する粒子からそれぞれのポリゴンに下ろした垂線の足はどのポリゴン上にもないため、この粒子には力は働かない。この粒子に最も近い位置に存在するのはポリゴンの辺か頂点であるため、これらからの距離を計算すれば連続になる。しかしこの計算によって計算コストは大幅に増加するため、近似的にこのような接続部分においても粒子に働く力を連続にする手法を用いる。粒子の近傍に存在する n 個のポリゴンを $T = \{t_0, t_1, \dots, t_n\}$ として、まずポリゴン j の重心座標との距離を計算し、この値が最も小さいポリゴンを最も近くに存在するポリゴン $t_{closest}$ とする。ポリゴン j を構成する3頂点の座標 $\mathbf{v}_{j,0}, \mathbf{v}_{j,1}, \mathbf{v}_{j,2}$ を用いて、この条件は以下のように表すことができる。

$$t_{closest} = \arg \min_{t_j \in T} \left(\mathbf{x} - \frac{\mathbf{v}_{j,0} + \mathbf{v}_{j,1} + \mathbf{v}_{j,2}}{3} \right)^2 \quad (5)$$

ここで3頂点は同じ質量を持つものとし、重心座標は3頂点座標それぞれに1/3の重みを付けて足し合わせたものとなる。

そしてポリゴン $t_{closest}$ との距離 d_t をこのポリゴンを構成する3頂点の座標を用いて以下のように求め、布と流体粒子までの距離とする。

$$d_t = |(\mathbf{v}_{j,1} - \mathbf{v}_{j,0}) \times (\mathbf{v}_{j,2} - \mathbf{v}_{j,0}) \cdot (\mathbf{x} - \mathbf{v}_{j,0})| \quad (6)$$

このように計算することでポリゴンの接続部分でも力を連続にすることができ、求まった粒子と布との距離が布の厚さ ϵ よりも小さければ布と粒子は衝突していると判定する。本研究では布の厚さ ϵ は粒子間平均距離とした。ポリゴンの接合部以外でも布の境界部分で力が不連続になる。しかし、この不連続性は計算結果に視覚的に大きな影響を与えなかったため、特別な処理は行なわなかった。

5.3 流体の衝突応答

流体の圧力項は粒子数密度を一定にするように働く力であり、粒子間距離が初期粒子間距離よりも近い場合には初期粒子間距離に戻す力となる。布と流体粒子間の力もこのようにモデル化する。つまり流体粒子 i と布との距離 d が ϵ より小さいときには、距離を ϵ に戻す力を加える。流体粒子が衝突しているポリゴンの法線ベクトル \mathbf{n} を用いて反発力 \mathbf{f}_i^{press} は以下のように計算する。

$$\mathbf{f}_i^{press} = m_i \frac{(\epsilon - d)\mathbf{n}}{dt^2} \quad (7)$$

また流体計算の影響半径内に布が存在する場合は粘性力を計算する。粒子座標から布に垂線を下ろし、その垂線に垂直な平面上に均一に粒子が並んでいると仮定する。そしてそれらの粒子が全て同じ速度 \mathbf{v} と密度 ρ を持つとして以下のように布からの粘性力を計算する。

$$\mathbf{f}_{i,wall}^{vis} = -\mu \frac{m}{\rho} (\mathbf{v} - \mathbf{v}_i) \sum_{j \in Wall} \nabla W_{vis}(\mathbf{r}_{ij}) \quad (8)$$

布上で粒子は均一に並んでいると仮定しているため、それらの粒子の配置は流体粒子 i と布までの距離によって一意に決まる。そのため、重み関数の和はあらかじめ計算しておき、そのテーブルから値を参照することで計算することができる。

流体粒子が布と近づくときに流体粒子の密度が上昇するが、布上に粒子が存在しないので布に近づいた流体粒子の密度が下がってしまう。これを防ぐために、布による密度の寄与を流体とは別に計算する。ここでも粘性項と同じように粒子座標から布に垂線を下ろし、その垂線に垂直な平面上に均一に粒子が並んでいると仮定して、以下のように布による密度の寄与 $\rho_{i,wall}$ を計算する。

$$\rho_{i,wall} = m \sum_{j \in Wall} W(\mathbf{r}_{ij}) \quad (9)$$

布上に均一に粒子が存在していると仮定するとそれらによる密度への寄与の重み関数の和もあらかじめ計算することができる。

5.4 布の衝突応答

布に働く力は流体に働く力の反力である。よってあるポリゴン i に働く力 \mathbf{F}_i はそのポリゴンに衝突した流体粒子に働く力 \mathbf{f}_j を用いて求めることができる。ポリゴン i に働く力 \mathbf{F}_i は以下のように求めることができる。

$$\mathbf{F}_i = - \sum_{j \in \text{Colliding}} \mathbf{f}_j \quad (10)$$

この力は厳密には流体粒子に働く力の作用点と力の方向を考慮しなければならないが、ここではそれらの力は面の法線方向と平行であり、全てポリゴンの重心座標に働くことと近似した。するとポリゴンを構成する3頂点に働く力 $\mathbf{f}_{i,0}, \mathbf{f}_{i,1}, \mathbf{f}_{i,2}$ は、3頂点に同じように分配されるので

$$\mathbf{f}_{i,0} = \mathbf{f}_{i,1} = \mathbf{f}_{i,2} = \frac{\mathbf{F}_i}{3} \quad (11)$$

と計算することができる。

6. GPU を用いた高速化

6.1 データ構造

GPU 上で計算を行なうために、流体と布の物理量を全てテクスチャに格納する必要がある。流体粒子の物理量は座標、速度、密度なのでこれらを格納するテクスチャを用意し、テクスチャのテクセル1個に流体粒子1個の物理量を格納する。布は粒子の座標、粒子の速度、粒子の接続情報が必要であるのでこれらをテクスチャに格納する。また流体の計算を行なうときにはボクセルの1辺の長さが粒子径の格子が必要である。しかしGPUは3次元テクスチャに書き込みを行なうことができないので、3次元格子を2次元格子に分割し、1枚の2次元テクスチャに敷き詰めることで表現する¹⁰⁾。布と流体の相互作用の計算においても布のポリゴンを格納する格子が必要なのでこの格子も同様に2次元テクスチャで表現する。

6.2 流体計算

流体計算はまず粒子座標での密度を計算し、それを用いて圧力項を計算する。そして粘性項と外力項を計算し速度を更新し、その速度を用いて流体座標を更新する。各流体粒子にはテクスチャ座標が1つ割り当てられ、座標、速度、密度テクスチャ内のその座標に物理量が格納されている。そのため、近傍粒子がわかればそのテクスチャ座標にあるそれぞれのテクスチャの物理量を読み出し、他のテクスチャのその座標に出力することで1タイムステップの計算を行なうことができる。近傍粒子探索を効率的に行なうにはまず格子を構築する必要がある。流体粒子の格子の構築は Harada



図1 Results of a simulation in which a fluid is poured onto a sheet of cloth.

らが開発した手法を用いて行なった⁹⁾。

6.3 布の格子の構築

布のポリゴンの番号を格納する格子を構築するために、まずはポリゴンの重心座標を計算する。そしてそれぞれの重心に頂点を1個割り当て、その頂点を格子内での重心の座標に動かし、流体粒子の格子を構築するのに用いた方法と同様な方法を用いて行なった⁹⁾。粒子番号のかわりにポリゴン番号を色と深度として出力した。

6.4 相互作用の計算

圧力項は複数段階に分けて計算する。まずは流体粒子と布との衝突判定を行なうため、作成した布の格子を用いてそれぞれの粒子から最小距離に存在する布ポリゴンの番号と距離を求める。この処理はそれぞれの粒子の格子内での座標を計算し、そのボクセルと近傍のボクセルに格納されているポリゴンの番号を求めることで行なうことができる。そしてこの番号を用いてポリゴンを構成する3頂点の座標を読み出し、5.2で述べた手法を用いて布と流体粒子までの距離を求める。このときに最近距離に存在するポリゴンの番号も出力する。布と流体粒子との距離を計算することができたので、その距離が ϵ より小さい場合には式(7)を用いて流体粒子に働く圧力項を計算する。

次に布のポリゴンに働く力を計算する。この計算では粒子に働く力の計算と同様にポリゴンの周辺に存在する粒子座標を探して力を計算することも可能であるが、本研究で用いたポリゴンは流体粒子よりも大きい。そのため、多数の粒子と距離計算を行なわなければならない。しかし流体粒子に働く力とその粒子と最小距離に存在するポリゴンの番号が分かっているため、ポリゴンに働く力は粒子に働く力の和を取ることで近傍粒子を探索せずに計算することができる。布を構成する頂点に働く力は2次元テクスチャ上のテクセル1個ずつに格納される。また流体粒子に働く力も2次元テクスチャ上のテクセル1個ずつに格納されている。この処理は1対1の写像ではなく、粒子に働く力を分散させる処理であるため、フラグメントシェーダでは処理することができない。そこでパーテックスシェーダで流体粒子に働く力のテクセルを読み出し、その力を

対応する布の粒子の力を保持するテクスチャのピクセルに1ピクセルの点として出力することによって計算することができる。流体粒子に働く力の反力は布のポリゴンに働くため、そのポリゴンを構成する3粒子のピクセルに力を出力しなければならない。この処理は流体粒子1個に3頂点を割り当て、それぞれ3個の粒子のピクセルに力を出力するようにして処理する。またポリゴンを構成する頂点は各ポリゴン間にて共有されているため、ある頂点に働く力は共有されているポリゴン全てに働く力の和となる。そこでアルファブレンディングを用いてそれらの和を計算する処理を行った。

5.3で述べたように粘性項と密度への布の寄与の計算において重み関数の和はあらかじめ計算しておくことができる。影響半径内の数点において値を計算し、それ以外の値は計算した点の値を線形補間することで求めた。そして計算した点での値を1次元テクスチャに格納してGPUに送った²⁰⁾。流体の密度と粘性項の計算では求めた布までの距離を用いて布の寄与を計算した。

7. 結 果

本手法を Core 2 X6800 CPU, 2.0GB のメモリ, GeForce 8800GTX を搭載した PC 上で実装した。プログラムは C++, OpenGL, C for Graphics を用いて開発した⁴⁾。本章で示す結果はそれぞれの流体粒子に濃度分布を与え、それらの和を取り陰関数曲面を構築した。そしてそれらから流体の表面を Marching Cubes を用いて抽出し、レイトレーシングを行なった¹⁴⁾。

図1に示すシーンは1枚の布の上に流体を流入させた結果である。流体を流入させているときには圧力によって布が変形し大きくたわんでいるが、流入が終わるとたわみが解消されている。図2は1枚の布に複数の液滴を落下させた計算結果である。液滴が布に接触すると布が形を変えているのがわかる。また2枚の布を用いた結果を図3に示す。これらの計算では全て65,536個の流体粒子を用いており、1枚の布は8,192枚のポリゴンから構成されている。これらの計算のフレームレートと流体粒子数を16,384個に減らしたシミュレーションのフレームレートを表1に示す。また布との相互作用を計算せずに流体だけの計算を行なった場合の結果も表1に示しており、16,384粒子を用いた場合の計算が約16fps、65,536粒子を用いた場合が約64fpsであった。この結果よりカップリングの計算コストは流体計算に比べ低いことがわかる。本手法は

表1 Number of fluid particles and frame rates.

	16,384	65,536
Figure 1	53.3	14.9
Figure 2	49.2	13.1
Figure 3	45.7	12.3
Fluid	64.1	16.6



図2 Results of a simulation in which droplets are dropped onto a sheet of cloth.

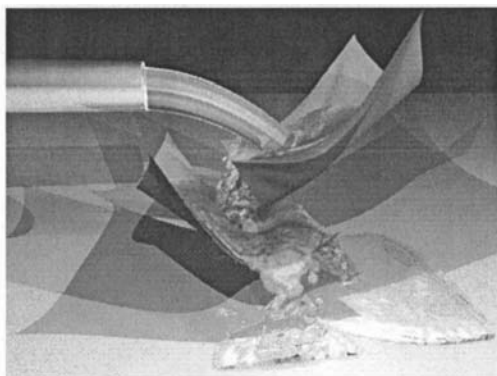


図3 Results of a simulation in which a fluid is poured onto two sheets of cloth.

布同士の衝突や布の自己衝突を計算していないため、それらを計算することはできない。布同士の衝突等は今後の研究課題の1つである。

8. 結 論

本研究ではSPHを用いて流体と布との相互作用をリアルタイムで計算する手法を提案した。流体粒子に働く力を空間で連続にする手法も提案した。流体と布の相互作用の計算を効率化するために流体粒子と布の座標を格納する2個の格子を導入した。また本手法は完全に並列化することができることを示し、GPUを用いて高速化を行なった。

本論文で示した粒子と布との相互作用計算モデルは流体以外の粒子ベースシミュレーション手法にも用いることが可能である。Distinct Element Methodによる粉体シミュレーションや、粒子を用いて剛体を表現する剛体シミュレーションともカップリングすることが可能である⁹⁾²¹⁾。本研究では粒子とポリゴンとの相

互作用を GPU 上で行ない高速化したが、四面体要素から成る弾性体のシミュレーションにも手法を応用することができる。

今後は複数枚の GPU を用いてより一層の高速化を行ない、リアルタイムシミュレーションの表現力の向上を行なっていく (図 4)。

参 考 文 献

- 1) T. Amada, M. Imura, Y. Yasumoto, Y. Yamabe, and K. Chihara. Particle-based fluid simulation on gpu. In *2004 ACM Workshop on General-Purpose Computing on Graphics Processors*, 2004.
- 2) N. Chentanez, T. G. Goktekin, B. E. Feldman, and J. F. O'Brien. Simultaneous coupling of fluids and deformable bodies. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 83–89, 2006.
- 3) P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, Vol. 29, pp. 47–65, 1979.
- 4) R. Fernando and M. Kilgard. *The Cg Tutorial*. Addison-Wesley Pearson Education, 2003.
- 5) O. G enevaux, A. Habibi, and J. M. Dischler. Simulating fluid-solid interaction. In *Graphics Interface*, pp. 31–38, 2003.
- 6) E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. Coupling water and smoke to thin deformable and rigid shells. *ACM Transactions on Graphics*, Vol. 24, pp. 910–914, 2005.
- 7) T. Harada. *GPU Gems 3*, chapter Real-time Rigid Body Simulation on GPUs. Addison-Wesley Pearson Education, 2007.
- 8) T. Harada, S. Koshizuka, and Y. Kawaguchi. Smoothed particle hydrodynamics on gpus. In *Proc. of Computer Graphics International*, pp. 63–70, 2007.
- 9) T. Harada, M. Tanaka, S. Koshizuka, and Y. Kawaguchi. Acceleration of distinct element method using graphics hardware. *Transactions of JSCEs*, Vol. 20070011, , 2007.
- 10) M. J. Harris, W. V. Baxter, T. Scheuermann, and A. Lastra. Simulation of cloud dynamics on graphics hardware. In *Proc. of the SIGGRAPH / Eurographics Workshop on Graphics Hardware*, pp. 92–101, 2003.
- 11) P. Kipfer and R. Westermann. Realistic and interactive simulation of rivers. In *Proc. of the 2006 Conference on Graphics Interface*, Vol. 137, pp. 41–48, 2006.
- 12) A. Kolb and N. Cuntz. Dynamic particle coupling for gpu-based fluid simulation. In *Proc. of 18th Symposium on Simulation Technique*,

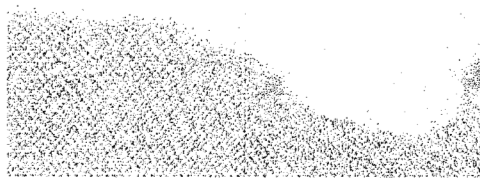


図 4 Real-time simulation on multiple GPUs. Two GPUs are used for simulation and one GPU is used for rendering.

- pp. 722–727, 2005.
- 13) S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nucl. Sci. Eng.*, Vol. 123, pp. 421–434, 1996.
- 14) W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proc. of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 163–169, 1987.
- 15) B. K. Mishra. A review of computer simulation of tumbling mills by the discrete element method: Parti-contact mechanics. *International Journal of Mineral Processing*, Vol. 71, No. 1, pp. 73–93, 2003.
- 16) M. M uller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proc. of SIGGRAPH Symposium on Computer Animation*, pp. 154–159, 2003.
- 17) M. M uller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. Interaction of fluids with deformable solids. *Journal of Computer Animation and Virtual Worlds*, Vol. 15, No. 3, pp. 159–171, 2004.
- 18) M. M uller, B. Solenthaler, R. Keiser, and M. Gross. Particle-based fluid-fluid interaction. In *Proc. of Siggraph Symposium on Computer Animation*, pp. 237–244, 2005.
- 19) John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Kr uger, Aaron E. Lefohn, and Timothy J. Purcell. A survey of general-purpose computation on graphics hardware. *Eurographics 2005, State of the Art Reports*, pp. 21–51, 2005.
- 20) 原田隆宏, 越塚誠一. Sph 法における壁境界計算手法の改良. *情報処理学会論文誌*, Vol. 48, No. 4, pp. 1838–1846, 2006.
- 21) 原田隆宏, 田中正幸, 越塚誠一, 河口洋一郎. Gpu を用いたリアルタイム剛体シミュレーション. *情報処理学会グラフィックスと CAD 研究会*, 2007–CG–126, pp. 79–84, 2007.