

粒子ベースシミュレーションの並列化

原田 隆 宏[†] 田 中 正 幸[†]
越 塚 誠 一[†] 河 口 洋 一 郎[†]

現在のプロセッサは並列アーキテクチャに向かっている。そのため、その計算機性能を十分に発揮させるためにデータ並列処理を行うアルゴリズムを開発する必要がある。本論文では並列化されたリアルタイム粒子ベースシミュレーション手法を提案する。粒子ベースシミュレーションを Graphics Processing Units (GPUs) を並列計算機として用いて高速化する。並列化したことによって、リアルタイムシミュレーション今まで計算可能だった粒子数の十倍以上の粒子数を計算できるようになった。本論文は粒子ベースシミュレーションにおいて流体と剛体を相互作用させる手法も提案する。この手法の利点は計算コストが低だけでなく、そのアルゴリズムを並列化できることにもある。流体、剛体、そしてそれらの相互作用をリアルタイムで計算することによって、並列化された粒子ベースシミュレーションの能力を示す。

Parallelization of Particle-based Simulations

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

Since processors are shifting towards parallel architecture at the present time, it is important to develop data-parallel algorithms to exploit their computational power. This paper presents a parallelized real-time particle-based simulation method. Particle-based simulation is accelerated by using Graphics Processing Units (GPUs) as a parallel computation platform. With parallelized particle-based simulation, the number of particles which can be simulated in real-time is increased by more than an order of magnitude. This paper also presents a particle-based method to interact fluids and rigid bodies. Rigid bodies are represented by a set of particles. The benefits of this method are not only low computational cost but also parallelism of its algorithm. By simulating fluids, rigid bodies and their interactions in real-time, we demonstrate the capability of the parallelized particle-based simulation.

1. 序 論

我々の身近に存在するものの大半は物理法則に従っている。たとえば流体の複雑な挙動は手で作成するのは困難だが、支配方程式を解くことによって求めることができるため、コンピュータグラフィックスにおいて物理シミュレーションは重要である。近年、流体をはじめ剛体、弾性体などのシミュレーションについて様々な研究が行われてきた。映像制作において求められる物理シミュレーションは正確さではなく、計算速度や安定性に重点が置かれるため、高速なシミュレーション手法の開発はコンピュータグラフィックスにおいて重要な研究課題の1つである。映像制作以外においても、ゲームなどのリアルタイムアプリケーション

でも物理シミュレーションを用いることができ、これらに応用するためにはリアルタイムに計算しなければならない。

また現在のプロセッサテクノロジーは周波数を向上させるのではなく、効率性を向上させる方向に向かっている。並列アーキテクチャを備えたプロセッサも多く存在する。Cell Broadband Engine (Cell BE) Architecture は汎用計算用に開発されたマルチコアプロセッサであり、Graphics Processing Units (GPUs) はグラフィックス処理に特化した並列プロセッサである。さらに CPU もマルチコア化されてきている。これらの計算能力を最大限に活用することができる並列計算アルゴリズムを開発していかなければならない。

粒子ベースシミュレーションはメッシュを用いない計算手法である。粒子間で接続情報を持たないため、近傍粒子を毎タイムステップで探索しなければならない。視覚的に十分な結果を得るためには、多くの粒子

[†] 東京大学

The University of Tokyo



図 1 グラスと流体のリアルタイムシミュレーション．積み上げられたグラスを流体で満たし、グラスを投げ込んだ

Fig. 1 Real-time simulation of glasses and liquid. Glass tower is filled with liquid and a glass is thrown onto the scene

を用いなければならないが、多くの粒子の挙動をリアルタイムで計算するには計算コストが高かった．既存の GPU を並列計算機としてシミュレーションに用いた研究はすべて計算ノードの接続関係が計算中に変化しないという共通点を持っていた．たとえば格子を用いた流体計算では各計算点はその計算点を囲む点のみと相互作用し、布のシミュレーションでは質点どうしの接続は計算中に変化しない^{13),24)}．しかし粒子ベースシミュレーションは、計算粒子は自由に動くことが可能であり、タイムステップごとに相互作用する計算粒子が変化する．そのため、計算粒子の近傍粒子を探索しなくてはならない．本研究では計算粒子の近傍粒子探索を並列化し、計算ノードの接続関係が動的に変化する粒子ベースシミュレーションも並列化することが可能であり、並列計算機上で計算することができることを示す．本研究では GPU を並列計算機として用いた．その結果、スカルプロセッサ 1 個では得ることのできないパフォーマンスを得ることができた．本論文では粒子ベースシミュレーションで流体と剛体の相互作用を計算する手法も提案する．この手法では剛体は粒子で表され、計算コストが低く、並列処理可能であるという利点がある．本手法を実装し、図 1 をはじめとする様々なシーンの計算を行い、手法の検証を行った．

2. 関連研究

Foster and Metaxas がコンピュータグラフィックスの分野では初めて 3 次元ナビエーストックス方程式を解いて流体計算を行い¹⁷⁾、Stam が Semi-Lagrangian 法を CG 分野に導入した⁴⁷⁾．Foster and Fedkiw は自由表面追跡に Level Set 法を導入した¹⁶⁾．その後、界面追跡手法の改良¹⁵⁾、粘弾性流体¹⁹⁾、融解⁹⁾、混相流³⁶⁾、2 次元と 3 次元のカップリング²⁷⁾、Octree³⁵⁾、四面体メッシュを用いた計算³⁰⁾ などの研究が行われて

きた．また流体計算手法はこれらの格子を用いた計算手法のほかに格子を使わず粒子を用いる Lagrangian の手法がある．Premoze らは Moving Particle Semi-Implicit (MPS) Method を⁴⁴⁾、Cani and Desbrun は Smoothed Particle Hydrodynamics (SPH) をコンピュータグラフィックスに応用した⁸⁾．Lagrangian 手法は移流によって質量損失が起こらないため、リアルタイムシミュレーションに向いていることを Müller らは示し⁴⁰⁾、ほかに混相流⁴²⁾、相転移⁵⁰⁾の研究も行った．Kipfer and Westermann は疎な粒子分布の場合に適したデータ構造を用いて SPH を高速化した²⁹⁾．自由表面流れのシミュレーションは計算コストが高く、リアルタイムアプリケーションに応用するためにはまだ研究の余地がある．

剛体シミュレーションに関しても多く研究が行われてきた．Baraff は剛体間の接触の際に働く力を解析的に求める手法を開発し²⁾、剛体の接触に関して研究を行った³⁾⁻⁶⁾．剛体の衝突に力積ベースの手法を用いた研究も行われた^{22),37),45)}．Mirtich は衝突に関係している剛体のみを衝突時刻に戻す Timewarp アルゴリズムを開発した³⁷⁾．パネとダンパを用いるペナルティ法を衝突応答に用いた研究もある^{39),46)}．Hasegawa らはこのペナルティ法を用いてリアルタイムシミュレーションを行っているが、剛体数は非常に少なく、凸形状のみしか扱えないという問題がある^{25),26)}．Guendelman らは形状表現にポリゴンと符号付き距離関数を用いて非凸形状の剛体に関して衝突計算を行った²⁰⁾．Kaufman らは多数の剛体間の接触を高速に解く手法を開発した．しかし剛体シミュレーションではまず衝突検出を行わなければならない、剛体シミュレーション自体を高速化するためには、高速な衝突検出手法が必要である²⁸⁾．Dingliana らは剛体を階層構造を持つ球で表現し衝突計算を行う手法を開発し 2 次元計算を行い¹⁴⁾、Suzuki らはこの手法を 3 次元に拡張した⁴⁸⁾．Bell らは剛体

を複数の大きさの異なる球で表現して衝突計算を行った⁷⁾。このように多く剛体計算の研究が行われてきたが、リアルタイムで多数の剛体を用いたシミュレーションを行った研究はない。

剛体と流体のカップリングの研究も行われてきた。Yngveらは圧縮性流体を用いた爆発の計算を行い、圧縮性流体と剛体とのカップリングを行った⁵¹⁾。Génevauxらは剛体を粒子をバネでつないだ弾性体で表し、流体のマーカー粒子と剛体粒子間の力を計算することによって流体と剛体のカップリングを行った¹⁸⁾。Carlsonらは剛体を流体として解き、剛体に変形の拘束条件を課すことで剛体と流体のカップリングを行った¹⁰⁾。Takahashiらは流体と飛沫と泡の計算を行い、さらに剛体と流体のカップリングも行った⁴⁹⁾。TakahashiらはVolume of Solidという各ボクセルでの剛体の占める割合を導入して剛体部分を特定した。そして剛体境界部分に圧力計算の境界条件を課し、計算された圧力を用いて剛体にかかる力を計算した。MüllerらはSPHで流体を解き、Lagrangianメッシュで弾性体の計算を行い、弾性体メッシュ表面に仮想粒子群を毎ステップ生成して流体粒子とそれらの間で相互作用を計算し、数千粒子でリアルタイムシミュレーションを行った⁴¹⁾。Guendelmanらは薄膜と流体の相互作用を弱連成で解いた²¹⁾。Klingnerらは境界適合格子を用いて、流体の圧力のポワソン方程式に剛体の圧力境界条件を組み込み、強連成で相互作用の計算を行った³¹⁾。ChentanezらはKlingnerらの手法を拡張し弾性体と流体をカップリングさせた¹¹⁾。自由表面流れ流体シミュレーションと剛体シミュレーションをカップリングする研究は行われているが、それらをリアルタイムで計算するという研究はほとんど行われていない。

GPUは処理によってはCPUよりも高速な計算が可能であり、様々な研究が行われてきた⁴³⁾。AmadaらはSPHをGPUで高速化したが、すべての計算をGPUで行うことはできず、近傍粒子探索はCPUで行った¹⁾。Kolbらの手法では粒子位置における物理量を求める際にまず格子上の値を求め、それらを補間して求めるため補間による数値拡散を引き起こしてしまう³²⁾。

3. 流体シミュレーション

3.1 支配方程式

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。

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

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

ρ , \mathbf{U} , P , ν , \mathbf{g} はそれぞれ流体の密度, 圧力, 速度, 動粘性係数そして重力である。

3.2 Smoothed Particle Hydrodynamics

SPHでは座標 \mathbf{x} での物理量は近傍粒子の持つ物理量の重み付け和で計算される。

$$\phi(\mathbf{x}) = \sum_j m_j \frac{\phi_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j) \quad (3)$$

m_j , ρ_j , \mathbf{x}_j は、それぞれ粒子 j の質量, 密度, 座標であり, W は重み関数である。

流体の密度は式 (3) を用いて近傍粒子の質量の重み付け和で求められる。

$$\begin{aligned} \rho(\mathbf{x}) &= \sum_j m_j \frac{\rho_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j) \\ &= \sum_j m_j W(\mathbf{x} - \mathbf{x}_j) \end{aligned} \quad (4)$$

また状態方程式より流体の圧力は

$$p = k(\rho - \rho_0) + p_0 \quad (5)$$

と計算される。 p_0 , ρ_0 はそれぞれ基準となる圧力と密度である。

式 (2) を粒子で計算するためには圧力項と粘性項を離散化する必要がある。対称性を考慮した粒子 i に働く圧力 $\mathbf{f}_i^{\text{press}}$ と粘性力 $\mathbf{f}_i^{\text{vis}}$ は近傍粒子の物理量を用いて以下のように計算される。

$$\mathbf{f}_i^{\text{press}} = - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W_{\text{press}}(\mathbf{r}_{ij}) \quad (6)$$

$$\mathbf{f}_i^{\text{vis}} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla W_{\text{vis}}(\mathbf{r}_{ij}) \quad (7)$$

また本研究では Müller らの用いた重み関数を用いた⁴⁰⁾。

4. 剛体シミュレーション

剛体の運動は並進運動と回転運動に分けて計算する。

4.1 並進運動

剛体に働く力 \mathbf{F} が剛体の重心の運動量 \mathbf{P} の時間微分である。

$$\frac{d\mathbf{P}}{dt} = \mathbf{F} \quad (8)$$

また剛体の速度 \mathbf{v} は重心の運動量 \mathbf{P} と剛体の質量 M を用いて以下のように求める。

$$\mathbf{v} = \frac{1}{M} \mathbf{P} \quad (9)$$

剛体の速度 \mathbf{v} を用いて剛体の重心位置 \mathbf{x} の時間微

分は

$$\frac{dx}{dt} = v \quad (10)$$

と表される．

4.2 回転運動

剛体に働く力 F は剛体の回転運動を生み、角運動量を変化させる．角運動量 L の時間微分は

$$\frac{dL}{dt} = r \times F \quad (11)$$

と表される．角速度 w は角運動量 P と時刻 t での慣性テンソル $I(t)$ を用いて

$$w = I(t)^{-1}P \quad (12)$$

と計算される．ここで時刻 t での慣性テンソルの逆行列 $I(t)^{-1}$ は、初期状態での慣性テンソル $I(t)$ と時刻 t での回転行列 $R(t)$ を用いて

$$I(t)^{-1} = R(t)I(0)^{-1}R(t)^{-1} \quad (13)$$

と計算する．そして式 (12) で求めた角運動量 w を用いて、クォータニオンの変化量 dq を以下のように計算する．

$$dq = \left[\cos\left(\frac{\theta}{2}\right), a \sin\left(\frac{\theta}{2}\right) \right] \quad (14)$$

ここで回転軸 a と回転角 θ は

$$a = \frac{w}{|w|} \quad (15)$$

$$\theta = |w|dt \quad (16)$$

と計算される．

計算されたクォータニオンの変化量と時刻 t でのクォータニオン $q(t)$ を用いて時刻 $t + dt$ でのクォータニオンを以下のように更新する．

$$q(t + dt) = dq \times q(t) \quad (17)$$

5. 相互作用

5.1 形状表現

本研究では剛体を均一な径を持つ粒子（ある大きさを持った球）の集合体として近似して相互作用を計算する．ポリゴンモデルから粒子データの生成は Harada and Koshizuka の手法を用いた²³⁾．剛体粒子を表す粒子の径を流体粒子の径と同じにすることで流体との相互作用の計算も容易に行えるようになる．この剛体を構成している粒子を剛体粒子と呼ぶ．図 2 にポリゴンモデルと粒子で表現した形状の例を示す．粒子の解像度を高くするほど物体の形状の近似は良くなり、解像度を低くするほど計算コストが下がるが、計算精度も同時に下がる．つまり粒子による表現は計算精度を変えることによって計算速度をコントロール可能なのでコンピュータグラフィックスでは有用である．

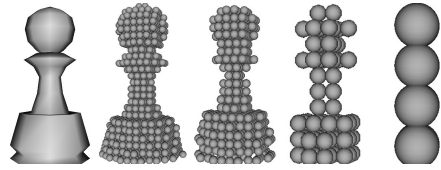


図 2 粒子による形状表現
Fig. 2 Particle representations.

5.2 剛体の衝突

剛体の衝突の検出は各剛体を構成している剛体粒子の衝突を検出することによって行う．そして衝突によって剛体に生じる力は剛体粒子に生じる力の和をとることで計算する．剛体粒子を導入することによって衝突検出を容易に行うことができるようになる．衝突検出が行われた後は、粉体の計算手法である個別要素法を応用して計算する¹²⁾．

5.2.1 反発力

個別要素法の計算では様々な反発力の計算方法が研究されているが、本研究では線形バネとダッシュポットを用いて反発力を計算する．このモデルでは衝突している粒子間にめり込み量に比例した反発力と相対速度に比例した減衰力を働かせる．2つの粒子 i と j がおり、これらの粒子の直径を d とするとそれらの距離 $|r_{ij}|$ が d より小さいときに、以下のバネによる力 f_s とダンパによる減衰力 f_d を働かせる．

$$f_s = -k(d - |r_{ij}|) \frac{r_{ij}}{|r_{ij}|} \quad (18)$$

$$f_d = \eta(v_j - v_i) \quad (19)$$

ここで k 、 η はバネ定数と減衰定数であり、これらと反発係数の関係を定式化することはできないため経験的に値を決定した．また $r_{ij} = r_j - r_i$ であり r_i 、 r_j は粒子 i 、 j の位置ベクトルである．

5.2.2 摩擦力

クーロンの法則によると動摩擦力は垂直応力に比例した力であり、相対運動方向に働く．よって粒子間の動摩擦力は

$$f_t = -\mu |f_n| \frac{v_t}{|v_t|} \quad (20)$$

と計算される．しかし式 (20) は剪断方向の相対速度 v_t が 0 のときは計算することができなくなる．そこで v_t がある値よりも小さい場合は動摩擦力は剪断方向の速度 v_t に比例した力として以下のように計算する．

$$f_t = k_t v_t \quad (21)$$

動摩擦力は静止摩擦力が最大値を超えた場合に働く．静止摩擦力は Lee and Herrmann が行っているように衝突した粒子のペアとその相対座標を保持しておくことで計算することができる³³⁾．しかし静止摩擦力を計

算しなくても視覚的に十分な挙動を計算することができたので、本研究では静止摩擦力は導入しなかった。

これらの反発力と摩擦力から球 i に働く衝突力 $\mathbf{f}_{i,rigid}$ は

$$\mathbf{f}_{i,rigid} = \mathbf{f}_s + \mathbf{f}_d + \mathbf{f}_t \quad (22)$$

と計算される。

6. 流体と剛体の相互作用

流体と剛体の相互作用は Carlson らの手法のようにまずは剛体を構成している粒子を流体として粒子にかかる力を計算する¹⁰⁾。剛体の粒子 i が流体から受ける力 $\mathbf{f}_{i,fluid}$ は圧力勾配による力と粘性力なので式 (6) と式 (7) を用いて

$$\mathbf{f}_{i,fluid} = \mathbf{f}_i^{press} + \mathbf{f}_i^{vis} \quad (23)$$

と計算される。

剛体を構成する粒子に働く力は剛体間の衝突による力 $\mathbf{f}_{i,rigid}$ と流体から受ける力 $\mathbf{f}_{i,fluid}$ であるので、剛体を構成する粒子にかかるこれらの力の和を計算することで、重心に働く力を計算することができる。

$$\mathbf{F}_c = \sum_{i \in RigidBody} (\mathbf{f}_{i,rigid} + \mathbf{f}_{i,fluid}) \quad (24)$$

また粒子に働く力と剛体の重心からの座標 \mathbf{r}'_i を用いて、その粒子に働く力が生む剛体のトルクを計算することができる。剛体全体に働く力は粒子にかかるトルクの和をとり

$$\mathbf{T}_c = \sum_{i \in RigidBody} \mathbf{r}'_i \times (\mathbf{f}_{i,rigid} + \mathbf{f}_{i,fluid}) \quad (25)$$

と計算できる。計算された力によって剛体の重心座標とクォータニオンを更新する。そして剛体を構成している粒子の座標をその粒子の初期配置、剛体の重心座標とクォータニオンから計算し、粒子の座標も更新する。

7. 近傍粒子探索

剛体間の衝突の計算も剛体と流体の相互作用の計算でも、各粒子の周囲に存在する粒子との距離から力を計算するため、近傍粒子を探索しなければならない。総当たりで計算すると粒子数 n の 2 乗のオーダの計算コストがかかり、粒子数が増加すればするほど計算コストは高くなってしまふ。効率的に近傍粒子を探索するために、計算領域を包括する計算格子 (グリッド) を導入する³⁸⁾。グリッドを構成する立方体要素がボクセルであり、各ボクセルにそれらが囲む計算領域を割り当てる。そしてボクセルには割り当てられた計算領域内に存在する粒子の番号を格納する。こうすることである粒子の周囲に存在する粒子はその粒子が格納さ

れているボクセルの周りのボクセル内に格納されている粒子に限定され、計算コストは粒子数 n のオーダになり効率良く探索することができる。本研究では 1 辺の長さが粒子の直径と同じ大きさのボクセルを用いた。すると粒子が最密構造の配置をとったとき、1 個のボクセルに入る粒子数は 4 個である。

8. GPU を用いた並列計算

今まで述べてきた流体と剛体シミュレーションとそれらの相互作用はすべての計算を並列化することが可能である。本研究では GPU を並列計算機として用いてシミュレーションを並列化させる方法を述べていく。本章ではまず GPU 上でのデータ構造について説明し、実装の詳細を述べていく。粒子からグリッドを構築する処理は流体と剛体のシミュレーションに共通の処理であるため、まず GPU 上でのグリッド構築手法について述べる。それから流体と剛体シミュレーションの計算について順に述べていく。

8.1 データ構造

GPU 上で物理計算を行うため、物理量はすべてテクスチャとしてビデオメモリ上に保持される。流体シミュレーションで各粒子が持つ物理量は座標と速度と密度であるため、それぞれテクスチャを用意する。計算を行うにあたり、流体の密度を計算しなければならないが、密度はそれぞれのタイムステップで粒子座標から計算されるため、1 枚のテクスチャを用意する。また剛体シミュレーションでは各剛体は座標、クォータニオン、並進運動量、角運動量であるため、それぞれにテクスチャを用意する。また慣性テンソルと初期粒子配置は同じ形状の剛体ならば共通のデータなので剛体ごとに用意する。流体と剛体シミュレーションに必要なこれらの物理量は計算要素 1 つに対して 1 ピクセルを割り当てる。つまり流体粒子 1 個と剛体 1 個に 1 ピクセルを割り当て、その物理量をピクセルに保持する。また本手法では近傍粒子探索を行うためにグリッドを構築するため、グリッドをメモリ上に確保しなければならない。計算領域を囲むグリッドは 3 次元格子であるが、GPU は 3 次元格子に出力することはできないため、Harris らが行ったように 2 次元格子を複数枚並べた 1 枚のテクスチャとしてメモリ上に確保する²⁴⁾。グリッドを構成するボクセル 1 個に 1 ピクセルを割り当てる。

8.2 グリッド構築

グリッド構築はボクセル内に存在する粒子番号を、対応するボクセルに格納する処理である。データを並列処理するとそれぞれのボクセルには最終的に 1 個

の値しか保持されないため、この処理では1個のボクセルに数個粒子が存在する場合には正しくグリッドを構築することができない。そこでこの処理を数段階に分けることによって、正しく処理することができるようになる。これはGPUを用いた処理においてもいえることであり、マルチパスレンダリングを行うことによってグリッドを正しく構築することができる。このデータ分配処理はGPUではそれぞれの粒子に対応付けられた頂点を用意し、それらに対応する粒子のグリッド内の座標に粒子番号を色として出力することで行うことができる。ここからマルチパスでグリッドを正しく構築する手法を述べていく。

あるボクセル b に格納される小さい順に並んだ粒子番号 i_0, i_1, i_2, i_3 を考える。そしてこの粒子番号をボクセル b に割り当てられたピクセルの Red, Green, Blue, Alpha (RGBA) チャンネルにそれぞれ格納することを目標として4パスで処理を行う。頂点は粒子番号の小さい順にレンダリングする。用いるパーテックスシェーダは頂点に対応する粒子座標に対応するグリッドテクスチャ内の位置に頂点を変位させ、フラグメントシェーダでは粒子番号を色と深度値として書き出す。

1パス目では i_0 を R チャンネルにレンダリングする。 i_0 に対応する頂点は最も小さい深度値を持つ頂点としてレンダリングされるため、デプスバッファを最大値で初期化し、小さい値がデプステストを合格するように設定することで i_0 を書き入れることができる。2パス目では i_1 を G チャンネルにレンダリングするが、このときにすでに値を書き入れた R チャンネルに上書きしないためにカラーマスクを行い R チャンネルには値を書き入れないようにする。1パス目で用いた深度バッファを用いるがデプステストは大きな値が合格するように設定することで i_0 は書き込まれなくなる。しかしデプステストだけだと最も深度値が大きい i_3 が最終的に書かれてしまう。 i_2, i_3 の書き込みを防ぐため、ステンシルテストを用いる。ステンシルファンクションでは書き込まれるごとにステンシル値を増加させるように設定し、ステンシルテストは1より大きい値を不合格にするように設定する。こうすることで各ピクセルには1回値がレンダリングされた後は書き込まれなくなる。つまり i_1 が書き込まれた後は値を書き込むことができなくなるの G チャンネルに選択的に i_1 を書き込むことができる。3パス目と4パス目では基本的に2パス目で行った処理と同じ処理を行う。しかし3パス目では RG チャンネル、4パス目では RGB チャンネルへの書き込みを防ぐため、カラーマスクを用いる。

そして深度バッファは初期化せずに使い続け、ステンシルバッファはそれぞれのパスごとに初期化する。

8.3 流体シミュレーション

8.3.1 密度と力の計算

流体の計算ではまず密度の計算を行う。この計算では近傍粒子を探索する必要がある。グリッドを参照することで粒子 i が格納されているボクセルの周辺のボクセルに格納されている粒子番号を得ることができる。これらの粒子番号から粒子座標を読み出し、式(4)を用いて密度を計算する。粒子に働く力を計算する際も近傍粒子を探索する必要があるが、この計算も密度の計算と同様にグリッドを参照することで近傍粒子番号を得ることができる。そしてその粒子番号から計算した密度、座標、速度を読み出し、圧力項と粘性項の計算を式(6)と式(7)を用いて行う。また剛体粒子に関しては衝突による力を計算する。

8.3.2 速度と座標の更新

計算された粒子に働く力を用いて速度と座標をオイラー陽解法で更新する。これらの処理は各粒子の持つ物理量のみを用いて計算することができるので、それぞれのテクスチャの各粒子に割り当てられたピクセルの値を参照するだけで行うことができる。

8.4 剛体シミュレーション

8.4.1 並進運動量と角運動量の更新

剛体の並進運動量の更新は式(8)の計算を行う必要があり、剛体を構成する粒子に働く力の総和をとる。また角運動量の更新は式(11)の計算を行う必要があり、この計算では粒子に働く力以外に、剛体の重心に対する粒子の相対位置ベクトル r' が必要なため、剛体のクォータニオンと重心座標と初期配置を用いて計算する。

8.4.2 座標とクォータニオンの更新

計算した速度を用いて式(10)を計算し、座標を更新する。また計算された角速度を用いて式(17)を計算し、クォータニオンを更新する。これらの計算は各剛体に割り当てられた物理量のみを用いた計算なので、それぞれの剛体に割り当てられたピクセルの値のみの参照で計算することができる。

9. 結 果

本手法を Core 2 X6800 CPU, 2.0 GB 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。本章で取り上げる計算結果はすべてリアルタイムで計算した結果であり、それらの計算結果をオフラインでレイトレーシングしたものである。流体



図 3 4,096 個のチェスの駒のリアルタイムシミュレーション
Fig. 3 Real-time simulation of 4,096 chess pieces.



図 4 2,731 個のトーラスのリアルタイムシミュレーション
Fig. 4 Real-time simulation of 2,731 toruses.

は計算粒子それぞれに濃度分布を与え、陰関数曲面を Marching Cubes を用いて抽出した³⁴⁾。

流体の計算に比べ剛体の計算は 2 つの理由から計算コストが低い。1 つ目の理由として、流体計算では近傍粒子探索を密度の計算と圧力項の計算において 2 回行わなければならないが、剛体計算では衝突計算で 1 回行っただけで済む。2 つ目の理由は衝突計算を行うとき、粒子 i が入っているボクセルと、そのボクセルに隣接しているボクセルに格納されている粒子番号を取り出さなければならない。しかし流体計算を行うときには影響半径が粒子径よりも大きいときには、それ以上のボクセル内に近傍粒子が存在している可能性があるため、より多くのボクセルから粒子番号を取り出さなければならない。

まず剛体計算のみの結果を示す。図 3 に 4,096 個のチェスの駒、図 4 に 2,731 個のトーラスを落としたシミュレーション結果を示す。フレームレートは表 1 に示すように 133.3 と 193.7 である。このフレームレートは 1 タイムステップ計算を行い、レンダリングも行ったものであるため、計算自体はより高速に行うことができる。これらの計算では 16,384 個の剛体粒子を用いている。両方の計算で用いた剛体粒子数が同じだがフレームレートが異なっているのはレンダリングしなければならない剛体数が多いことに起因して

表 1 剛体粒子数とフレームレート

Table 1 Number of rigid particles and frame rates.

	Number of particles	FPS
Figure 3	16,384	133.3
Figure 4	16,384	193.7

表 2 チェスの駒の数とフレームレート

Table 2 Number of chess pieces and frame rates.

Number of particles	FPS
256	636.9
1,024	355.9
4,096	133.3
16,384	40.7

表 3 CPU での計算時間 (time (a)) と GPU での計算時間 (time (b)) (ミリ秒)

Table 3 Simulation time on CPU (time (a)) and on GPU (time (b)) in milliseconds.

Number of chess pieces	Time(a)	Time(b)
1,024	12.5	1.41
4,096	51.8	2.81
16,384	195.1	7.96

いる。剛体数を変えてフレームレートを測定した結果を表 2 に示す。また剛体計算を CPU で実装して計算速度を計測し、本手法と比較したものを表 3 に示す。この結果はレンダリングを行わず計算のみを行ったものである。剛体数 16,384 において CPU での計算より 24.5 倍高速に計算できている。

次に流体と剛体の連成計算結果を示す。図 1 では 10 個のグラスを積み重ねたシーンに 2 カ所から流体を流入させたシミュレーション結果である。流体がグラスを上から満たしていった。そして流体流入後、1 個のグラスを投げ入れ、積み重なったグラスを崩れ落とさせた。剛体の相互作用では静止摩擦力の計算をせず、動摩擦力の計算しか行っていないが、このような複数の剛体が積み重なったシーンでも剛体をほぼ静止させることができた。次の計算例は密度のテストである。図 5 では流路が流体で満たされ、それぞれ異なった密度を持つ剛体を落とした。流体は圧力の均衡を保つため、高さを変えていることが分かる。図 6 では水槽を少量の水で満たし、256 個のチェスの駒を落とした。その後液滴を水槽に落とした。チェスの駒の密度として 2 種類の値を用いたため、重い駒は沈み、軽い駒は浮いている。図 7 には 1 個のグラスに水を注ぎ、グラスが水によって倒されるシミュレーションの結果を示す。図 8 では 1 軸で空中に固定された 5 個のギアが配置されたシーンにおいて 2 カ所から流体を流入させた。流体によってギアが回転し、その抵抗によ

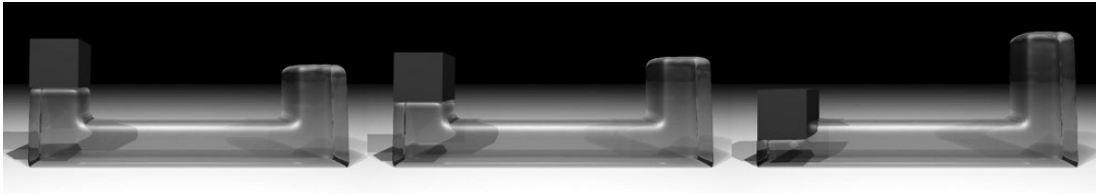


図 5 流路内にある流体と剛体．剛体の流体に対する比重はそれぞれ 0.5, 1, 2 である
Fig. 5 A liquid and a rigid body are in a channel. Relative densities of rigid bodies to the liquid are 0.5, 1 and 2, respectively.

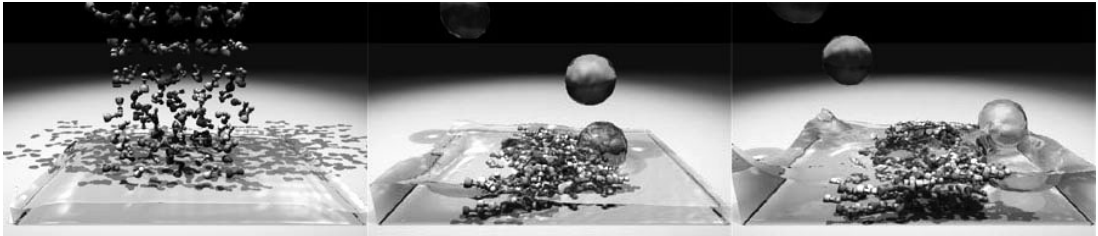


図 6 256 個のチェスの駒と液滴を水槽に落とした
Fig. 6 256 Chess pieces and balls of liquid are dropped into a tank.



図 7 1 個のグラスに流体を流入させた
Fig. 7 Liquid is poured into a glass.

り流れが乱れている様子が計算されている．図 9 に示すシミュレーションでは水をインタラクティブに操作し波を作り，その後，水面に船を浮かべた．この波によって船が倒される様子が示されている．これらの計算に用いた粒子数と 1 タイムステップの計算時間を表 4 に示す．約 6 万粒子を用いた流体と剛体の計算は約 17 fps で計算できている．

10. 結 論

本研究では粒子を用いたシミュレーションを並列化

しリアルタイムで計算する手法を提案した．GPU 上で粒子を用いたシミュレーションを並列化し，流体，剛体そしてそれらの相互作用をリアルタイムで計算した．本手法を用いることによって流体シミュレーションに関しては，従来リアルタイムシミュレーションで用いられていた粒子数の数十倍の計算をリアルタイムで行うことができた．また剛体計算では多数の剛体があるシーンをリアルタイムで計算した．またそれらの相互作用もリアルタイムで計算した．本研究ではリアルタイムシミュレーション結果のみ示したが，本手法を用いてオフラインシミュレーションの高速化も可能である．高解像度のシミュレーションを用いた映像制作において計算時間は重要なため，本手法は映像制作においても有用である．

本研究では SPH を用いた流体と剛体のシミュレーションを行ったが，本研究で示した手法は他の粒子法にも適用することが可能であるため，粒子を用いた弾性体やシェルの計算にも応用することができる．また本研究では流体のレンダリングはリアルタイムシミュレーション結果から表面抽出をオフラインで行ったが，表面抽出をリアルタイムで行う手法を開発し，シミュレーションとレンダリングをすべてリアルタイムで行うことのできる手法も研究していく．最後に本研究では並列計算のプラットフォームとして GPU を用いたが，本手法は他のプロセッサ上でも同様に並列計算することが可能であると考えられる．そこで今後マルチコアの CPU や Cell BE などを用いた並列化されたリアル

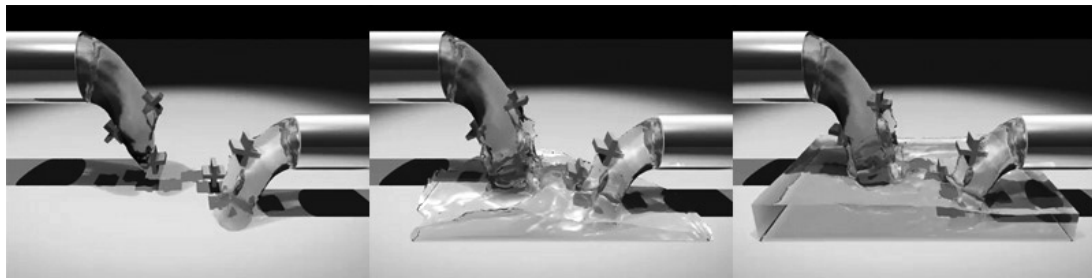


図 8 流体を水槽に流入させた。パドルは 1 軸によって空間に固定されている
Fig.8 Liquid is poured into a tank. Axes of paddles are fixed in the space.



図 9 波と船
Fig.9 A wave and ships.

表 4 流体粒子数とフレームレート
Table 4 Number of fluid particles and frame rates.

	Number of particles	FPS
Figure 1	49,153	17.1
Figure 5	8,193	91.3
Figure 6	64,512	16.2
Figure 7	12,289	75.2
Figure 8	64,512	15.2
Figure 9	245,760	3.85

タイムシミュレーションを研究していく。

参 考 文 献

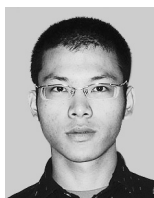
- 1) Amada, T., Imura, M., Yasumoto, Y., Yamabe, Y. and Chihara, K.: Particle-based fluid simulation on gpu, *2004 ACM Workshop on General-Purpose Computing on Graphics Processors* (2004).
- 2) Baraff, D.: Analytical methods for dynamic simulation of non-penetrating rigid bodies, *Computer Graphics*, Vol.23, No.3, pp.223-232 (1989).
- 3) Baraff, D.: Curved surfaces and coherence for non-penetrating rigid body simulation, *Computer Graphics*, Vol.24, No.4, pp.19-28 (1990).
- 4) Baraff, D.: Coping with friction for non-penetrating rigid body simulation, *Computer Graphics*, Vol.25, No.4, pp.31-40 (1991).
- 5) Baraff, D.: Fast contact force computation for nonpenetrating rigid bodies, *Computer Graphics Proc., Annual Conference Series*, pp.23-34 (1994).
- 6) Baraff, D.: Interactive simulation of solid rigid bodies, *IEEE Computer Graphics and Applications*, Vol.15, pp.63-75 (1995).
- 7) Bell, N., Yu, Y. and Mucha, P.J.: Particle-based simulation of granular materials, *Proc. 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.77-86 (2005).
- 8) Cani, M.P. and Desbrun, M.: Animation of deformable models using implicit surfaces, *IEEE Trans. Visualization and Computer Graphics*, Vol.3, No.1, pp.39-50 (1997).
- 9) Carlson, M., Mucha, P., Horn III, B.V. and Turk, G.: Melting and flowing, *Proc. ACM SIGGRAPH Symposium on Computer Animation*, pp.167-174 (2002).
- 10) Carlson, M., Mucha, P.J. and Turk, G.: Rigid fluid: Animating the interplay between rigid bodies and fluid, *ACM Trans. Graphics*, Vol.23, No.3, pp.377-384 (2004).
- 11) Chentanez, N., Goktekin, T.G., Feldman, B.E. and O'Brien, J.F.: Simultaneous coupling of fluids and deformable bodies, *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.83-89 (2006).
- 12) Cundall, P.A. and Strack, O.D.L.: A discrete numerical model for granular assemblies, *Geotechnique*, Vol.29, pp.47-65 (1979).
- 13) Cyril, Z.: Cloth simulation on the gpu, *ACM SIGGRAPH Sketches*, No.39 (2005).
- 14) Dingliana, J. and O'Sullivan, C.: Graceful degradation of collision handling in physically based animation, *Computer Graphics Forum*, Vol.19, No.3, pp.239-247 (2000).

- 15) Enright, D., Marschner, S. and Fedkiw, R.: Animation and rendering of complex water surfaces, *ACM Trans. Graphics*, Vol.21, pp.721–728 (2002).
- 16) Foster, N. and Fedkiw, R.: Practical animation of liquids, *Proc. SIGGRAPH*, pp.15–22 (2001).
- 17) Foster, N. and Metaxas, D.: Realistic animation of liquids, *Graphical Models and Image Processing*, Vol.58, No.5, pp.471–483 (1996).
- 18) G enevaux, O., Habibi, A. and Dischler, J.M.: Simulating fluid-solid interaction, *Graphics Interface*, pp.31–38 (2003).
- 19) Goktekin, T., Bargteil, A.W. and O'Brien, J.F.: A method for animating viscoelastic fluids, *ACM Trans. Graphics*, Vol.23, pp.464–467 (2004).
- 20) Guendelman, E., Bridson, R. and Fedkiw, R.: Nonconvex rigid bodies with stacking, *ACM Trans. Graphics*, Vol.22, pp.871–878 (2003).
- 21) Guendelman, E., Selle, A., Losasso, F. and Fedkiw, R.: Coupling water and smoke to thin deformable and rigid shells, *ACM Trans. Graphics*, Vol.24, pp.910–914 (2005).
- 22) Hahn, J.K.: Realistic animation of rigid bodies, *Computer Graphics*, Vol.22, No.4, pp.299–308 (1988).
- 23) Harada, T. and Koshizuka, S.: Fast solid voxelization using graphics hardware, *Transactions of Japan Society for Computational Engineering and Science* (2006).
- 24) Harris, M.J., Baxter, W.V., Scheuermann, T. and Lastra, A.: Simulation of cloud dynamics on graphics hardware, *Proc. SIGGRAPH/Eurographics Workshop on Graphics Hardware*, pp.92–101 (2003).
- 25) Hasegawa, S., Fujii, N., Koike, Y. and Sato, M.: Real-time rigid body simulation based on volumetric penalty method, *Proc. 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp.326–332 (2003).
- 26) Hasegawa, S. and Sato, M.: Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects, *Computer Graphics Forum*, Vol.24, No.3, pp.946–956 (2004).
- 27) Irving, G., Guendelman, E., Losasso, F. and Fedkiw, R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques, *ACM Trans. Graphics*, Vol.25, pp.812–819 (2006).
- 28) Kaufman, D.M., Edmunds, T. and Pai, D.K.: Fast frictional dynamics for rigid bodies, *ACM Trans. Graphics*, Vol.24, No.3, pp.946–956 (2005).
- 29) Kipfer, P. and Westermann, R.: Realistic and interactive simulation of rivers, *Proc. 2006 Conference on Graphics Interface*, Vol.137, pp.41–48 (2006).
- 30) Klingner, B.M., Feldman, B.E., Chentanez, N. and O'Brien, J.F.: Fluid animation with dynamic meshes, *ACM Trans. Graphics*, Vol.25, pp.820–825 (2006).
- 31) Klingner, B.M., Feldman, B.E., Chentanez, N. and O'Brien, J.: Fluid animation with dynamic meshes, *ACM Trans. Graphics*, Vol.25, No.3, pp.820–825 (2006).
- 32) Kolb, A. and Cuntz, N.: Dynamic particle coupling for gpu-based fluid simulation, *Proc. 18th Symposium on Simulation Technique*, pp.722–727 (2005).
- 33) Lee, J. and Herrmann, H.J.: Angle of repose and angle of marginal stability: molecular dynamics of granular particles, *Journal of Physics A*, Vol.26, pp.136–146 (1993).
- 34) Lorensen, W.E. and Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm, *Proc. 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp.163–169 (1987).
- 35) Losasso, F., Gibou, F. and Fedkiw, R.: Simulating water and smoke with an octree data structure, *ACM Trans. Graphics*, Vol.23, pp.457–462 (2004).
- 36) Losasso, F., Shinar, T., Selle, A. and Fedkiw, R.: Multiple interacting liquids, *ACM Trans. Graphics*, Vol.25, pp.812–819 (2006).
- 37) Mirtich, B.: Timewarp rigid body simulation, *Proc. SIGGRAPH 2000*, pp.193–200 (2000).
- 38) Mishra, B.K.: 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).
- 39) Moore, M. and Wilhelms, J.: Collision detection and response for computer animation, *Computer Graphics*, Vol.22, No.4, pp.289–298 (1988).
- 40) M uller, M., Charypar, D. and Gross, M.: Particle-based fluid simulation for interactive applications, *Proc. SIGGRAPH Symposium on Computer Animation*, pp.154–159 (2003).
- 41) M uller, M., Schirm, S., Teschner, M., Heidelberger, B. and Gross, M.: Interaction of fluids with deformable solids, *Journal of Computer Animation and Virtual Worlds*, Vol.15, No.3, pp.159–171 (2004).

- 42) Müller, M., Solenthaler, B., Keiser, R. and Gross, M.: Particle-based fluid-fluid interaction, *Proc. Siggraph Symposium on Computer Animation*, pp.237–244 (2005).
- 43) Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E. and Purcell, T.J.: A survey of general-purpose computation on graphics hardware, *Eurographics 2005, State of the Art Reports*, pp.21–51 (2005).
- 44) Premoze, S., Tasdizen, T., Bigler, J., Lefohn, A. and Whitaker, R.T.: Particle-based simulation of fluids, *Computer Graphics Forum*, Vol.22, No.3, pp.401–410 (2003).
- 45) Schmidl, H. and Milenkovic, V.J.: A fast impulsive contact suite for rigid body simulation, *IEEE Trans. Visualization and Computer Graphics*, Vol.10, No.2, pp.189–197 (2004).
- 46) Snyder, J.M., Woodbury, A.R., Fleischer, K., Currin, B. and Barr, A.H.: Interval methods for multi-point collisions between time-dependent curved surfaces, *Proc. 20th Annual Conference on Computer Graphics and Interactive Techniques*, pp.321–334 (1993).
- 47) Stam, J.: Stable fluids, *Proc. ACM SIGGRAPH 99*, pp.121–128 (1999).
- 48) Suzuki, K., Kubota, J. and Ohtsubo, H.: Rigid body simulation using volume based collision detection, *Structural Engineering/Earthquake Engineering*, Vol.22, pp.185–192 (2003).
- 49) Takahashi, T., Fujii, H., Kunimatsu, A., Hiwada, K., Saito, T., Tanaka, K. and Ueki, H.: Realistic animation of fluid with splash and foam, *Computer Graphics Forum*, Vol.22, No.3, pp.391–400 (2003).
- 50) Wicke, M., Hatt, P., Pauly, M., Müller, M. and Gross, M.: Versatile virtual materials using implicit connectivity, *Proc. Point-Based Graphics*, pp.73–82 (2006).
- 51) Yngve, G.D., O'Brien, J.F. and Hodgins, J.K.: Animating explosions, *Proc. ACM SIGGRAPH 2000*, pp.29–36 (2000).

(平成 19 年 1 月 26 日受付)

(平成 19 年 8 月 9 日採録)



原田 隆宏 (正会員)

昭和 56 年生。平成 18 年東京大学大学院工学系研究科システム量子工学専攻修士課程修了。同年東京大学大学院情報学環学際情報学府助手。平成 19 年 4 月助教。計算力学とコンピュータグラフィックスの研究に従事。日本機械学会、画像電子学会、ACM、Eurographics 各会員。



田中 正幸

昭和 57 年生。平成 19 年東京大学大学院工学系研究科システム量子工学専攻修士課程修了。同年株式会社東芝生産技術センター。粒子ベースのマルチフィジックスシミュレーションの研究に従事。日本機械学会会員。



越塚 誠一

昭和 37 年生。昭和 61 年東京大学大学院工学系研究科原子力工学専攻修士課程修了。同年東京大学工学部助手。平成 3 年博士 (工学)。同年講師。平成 5 年助教授。平成 16 年教授。連続体の力学シミュレーションの研究に従事。特に粒子法の開発を行う。平成 17 年に丸善より『粒子法』を出版。



河口洋一郎

昭和 27 年生。昭和 53 年東京教育大学大学院 (現、筑波大学大学院) 修了。平成 4 年筑波大学芸術学系助教授。平成 10 年東京大学大学院工学系研究科・人工物工学センター教授。平成 12 年東京大学大学院情報学環教授。昭和 52 年から CG に着手し、世界的 CG アーティストとして活躍中。第 100 回ベネチアビエンナーレ日本代表芸術家に選ばれる。