

陰関数曲面フィッティングを用いた パーティクル法における表面可視化手法の開発

Surface Extraction Method for Particle-based Simulation Using Implicit Function

竹谷健† Ken TAKETANI†, 藤澤誠† Makoto FUJISAWA†, 三河正彦† Masahiko MIKAWA† and 田中和世† Kazuyo TANAKA†

† 筑波大学 † University of Tsukuba

E-mail: †{taketani,fujis,mikawa,ktanaka}@slis.tsukuba.ac.jp

1 緒論

コンピュータグラフィックス (CG) で、自然現象を再現することは重要なテーマとなっている。特に、水の流れや炎といった流体现象のアニメーションは、映画やゲームなどの映像表現の分野において、今や欠かせないものとなっている。

流体シミュレーションの手法は、オイラー的手法とラグランジュ的手法に大別できる。オイラー的手法では、固定された座標系において各座標での物理量を時間とともに変化させていく。オイラー的手法の中で最も用いられるグリッド法では、計算空間を格子に分割し、それぞれの格子中心を計算点とし、速度、圧力などを時間ステップごとに更新していく。このとき、ナビエ・ストークス方程式と同時に、質量保存則から導かれる連続式も解かなければ、大きな質量の損失を起こす。結果、非常に大きな線形システムと解かなければならず、インタラクティブなシミュレーションには不向きである。

ラグランジュ的手法は、ある微小な流体塊に注目し、それを追跡しながらその物理量を変化させていく手法である。ラグランジュ的手法の代表例であるパーティクル法では、流体をパーティクルの集合とみなして、一定の質量をもったパーティクルを時間ステップごとに移動させていく。パーティクルそのものが流体の質量を表すため、パーティクル数が変わらなければ質量保存が満たされ、グリッド法のように連続式を解く必要がなく、計算コストが小さい。そのため、パーティクル法はよりインタラクティブなアプリケーションに応用できる可能性を有する。しかし、得られた結果を描画する際、パーティクルをそのまま描画するのではリアリティにかけてしまう。レンダリングにおいて重要となる光の反射、屈折は流体表面境界において起こる。そのため、パーティクルから、流体表面を生成しなければならない。

本論文では、粒子法の一つである SPH による液体シミュレーションの結果に対して、陰関数曲面フィッティングを用いて液体表面を可視化する手法の開発を目的とする。カーネル関数を用いたメタボール [1] による手法では

なく、ポイントベースレンダリング分野で用いられている陰関数曲面の点群へのフィッティング技術をパーティクル法に適用する。これにより、従来表現しきれなかった液体表面の特徴である波の先端の鋭い形状と滑らかな平面の両立が可能となる。

ポイントベースレンダリングで用いられるフィッティング手法は、数百万から数億もの大規模な点群を対象としており、リアルタイムのシミュレーションで使われる数万程度のパーティクルにそのまま適用することは難しい。本論文では、パーティクル法に特化したグループ分け、陰関数の合成法を提案する。これらの手法により、少ないパーティクルでも、流体の特徴をうまく再現できる手法を開発する。

2 関連研究

2.1 パーティクル法における表面可視化

パーティクル法において、表面を可視化手法するための様々な手法が開発されてきた。Müller ら [2] は、SPH でシミュレーションを行い、カーネル関数を用いたメタボールで液体表面形状を表現し、マーチングキューブ法 [?] により液体表面を抽出することで、インタラクティブな流体アニメーションを可能にした。この手法はパーティクルの濃度を使って表面を生成しているため、滑らかな面に凹凸が現れたり、波の先端のような鋭い形状に丸みが現れたりという問題がある。Yu ら [3] は、マーチングキューブに用いる陰関数を計算する際に異方性カーネルを用いることで滑らかな表面を生成した。また、Adams ら [4] は、シミュレーションにおいて、タイムステップごとに一定の条件でパーティクルを分割、合成することによりパーティクルの大きさを自由に変更できるようにした。これにより、表面からの距離・視点などの条件を考慮して適用的なレンダリングを行った。しかし、これらの手法もパーティクルの濃度を表面形状の生成に使っているため、同様の問題が発生する。

2.2 陰関数曲面による形状表現

3D スキャナ等を用いて取得する点群から表面を再構成するポイントベースモデリングの分野において、計測ノイズや欠損、不均一なサンプリング間隔をもつ点群データをそのままメッシュ頂点に使うことは不適切である。そこで、Lancasterら [5] は、点群に重み付き最小自乗法近似で陰関数曲面フィッティングを行った。また、Carrら [6] は、基底を用いて形状関数を定義する RBF(Radial Basis Functions) を開発した。Ohtakeら [7] は、点群を小領域に分け、それぞれ最小二乗法により形状関数を定義し、これらの関数を結合することで陰関数曲面を生成する MPU(Multi-level Partition of Unity) を開発した。さらに、Kanaiら [8] は点群を小領域の球で被覆し、各球体内で形状関数を定義し、それを GPU で直接レンダリングした。しかし、これらの手法は、密なデータへの適用であり、流体シミュレーションのような疎なデータに対して適用した研究はなされていない。本論文では、パーティクル法に特化したパーティクルのグループ分け、陰関数の合成法を提案する。これらの手法により、少ないパーティクルでも、流体の特徴をうまく再現できる手法を開発する。

3 提案手法

3.1 全体の流れ

提案手法では、まず液体のシミュレーションを SPH で行う。各ステップにおいて、SPH により液体を表すパーティクルの位置が決定される。そのパーティクルの位置からパーティクルの法線ベクトルを算出、流体表面付近にあるパーティクルを抽出する。さらに、フィッティングを容易にするため、表面パーティクルを表面の形状にあわせてグループ分けし、各グループで陰関数曲面フィッティングを行う。その後、グループ間の陰関数値の補間により、全体的な陰関数場をつくる。最後に、陰関数場からマーチングキューブ法で表面を生成する。

各ステップごとの、全体的なシミュレーションの流れは以下ようになる (図 1)。

1. SPH により粘性項, 圧力項, 外力項を計算し, パーティクルの速度, 位置を更新 (3.2 節)
2. 各パーティクルの法線ベクトルの算出 (3.3 節)
3. 表面パーティクルの抽出 (3.4 節)
4. 表面パーティクルを小区間ごとのグループに分割 (3.5 節)
5. グループごとに陰関数曲面フィッティング (3.6 節)
6. 陰関数場の生成とレンダリング (3.7 節)

これ以降、各処理について詳しく述べる。

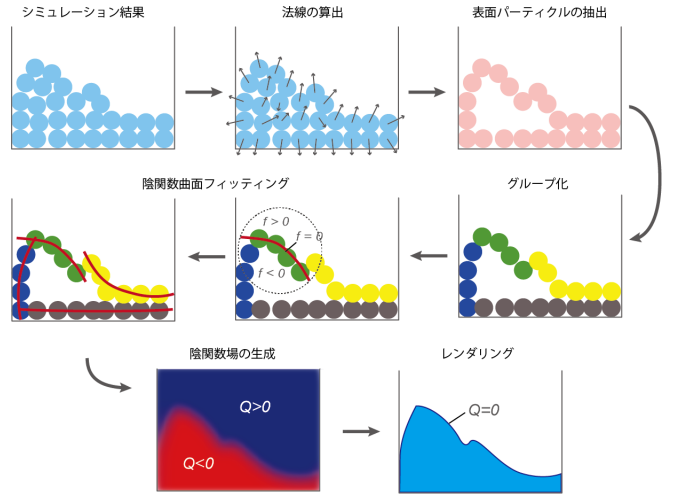


図 1: 全体の流れ

3.2 SPH による流れの計算

流体の支配方程式である非圧縮のナビエ・ストークス方程式を以下に示す。

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f} \quad (2)$$

ここで、 \mathbf{u} は流体速度、 μ は粘性係数、 ρ は流体の密度、 p は圧力、 \mathbf{f} は外力で重力など含まれる。

本論文では、液体のシミュレーションにラグランジュ的方法であるパーティクル法を用いている。パーティクル法はパーティクル自体が液体を表しているため、パーティクル質量が変化しないかぎり質量保存性が常に保持され (質量保存式である式 (1) を解く必要がない)、グリッド法において計算時間のかかる処理である液体表面の追跡の必要性がないことが利点である。また、粒子法では、グリッド法のような移流項の計算を行う必要がない。

SPH における座標 \mathbf{x} での物理量 ϕ は式 (3) で示すように周囲のパーティクル j のもつ物理量 ϕ_j の和で与えられる。

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

ここで、 m_j 、 ρ_j 、 \mathbf{x}_j はそれぞれパーティクル j の質量、密度、位置である。関数 W はカーネル関数である。

物理量 ϕ の勾配は、右辺第 1 項で使用し、

$$\nabla \phi(\mathbf{x}) = \sum_j m_j \frac{\phi_j}{\rho_j} \nabla W(\mathbf{x} - \mathbf{x}_j) \quad (4)$$

と表され、物理量 ϕ のラプラシアンは、右辺第 2 項で使用し、

$$\nabla^2 \phi(\mathbf{x}) = \sum_j m_j \frac{\phi_j}{\rho_j} \nabla^2 W(\mathbf{x} - \mathbf{x}_j) \quad (5)$$

と表される。\$W\$には、Müllerら[2]が提案したカーネルを採用した。密度、圧力項、粘性項について、それぞれPoly6カーネル、Spikyカーネル、Viscosityカーネルである。圧力\$p\$は、その液体の静止状態における密度を\$\rho_0\$として以下のような式となる。

$$p = k_{press}(\rho - \rho_0) \quad (6)$$

\$k_{press}\$は定数で、これ大きくすることで、ほぼ非圧縮な流体として計算できるが、計算が不安定になる。非圧縮性をSPHでも強制する方法にWCSPH[9]やPCISPH[10]があるが、これらは非常に小さなタイムステップ幅を要求するため、インタラクティブなアプリケーション向きではない。そのため、本論文では式(6)を用いることとする。

3.3 法線の算出

表面パーティクルの抽出やフィッティングの際に各パーティクルの法線ベクトルが必要になる。そこで近傍パーティクルとの相対位置ベクトルを用いて法線を算出する。

$$\mathbf{n} = (\mathbf{r}_i - \mathbf{r}_j) \sum_j w_n \quad (7)$$

ここで\$\mathbf{r}_i\$はパーティクル\$i\$の位置、\$\mathbf{r}_j\$は近傍パーティクルの位置である。また\$w_n(x)\$は距離に応じた重みであり、式(8)で与えられる。

$$w_n(x) = \frac{h}{|x|} - 1 \quad (8)$$

\$h\$はパーティクルが近傍パーティクルとして重み関数に寄与する有効半径である。

式(8)によって重み付けられた近傍パーティクルからの方向ベクトルの総和をそのパーティクルの法線ベクトルとする。図2にこの方法により法線を算出した結果を示す。表面の向きにベクトルが生成できている。

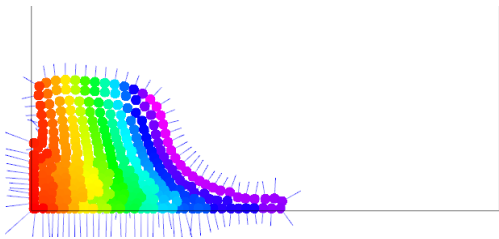


図 2: 法線の算出

3.4 表面パーティクルの抽出

本手法では、流体表面付近のパーティクル(以下、表面パーティクル)に対して陰関数曲面フィッティングを行

う。そのため全てのパーティクルの内、表面パーティクルのみを抽出する必要がある。そこで、下式に示すように、Solenthalerら[11]の近傍パーティクルへのベクトルの総和による推定を行い、表面パーティクルを抽出する。

$$\mathbf{d}_i = \frac{\sum_j (\mathbf{r}_i - \mathbf{r}_j) m_j}{\sum_j m_j} \quad (9)$$

ここで、\$\mathbf{d}_i\$はパーティクル\$i\$から近傍パーティクルへのベクトルの総和、\$m_j\$はパーティクルの質量、\$\mathbf{r}_i\$は任意のパーティクルの位置、\$\mathbf{r}_j\$は近傍パーティクルの位置である。

任意のパーティクルが表面付近にある場合、そのパーティクルから近傍パーティクルへのベクトルは一定方向に集中することから、\$\mathbf{d}_i\$の絶対値は大きくなる。一方、表面付近にないパーティクルでは、全周囲に近傍パーティクルがあるため、ベクトルが分散し、\$\mathbf{d}_i\$の絶対値は小さくなる。したがって、\$\mathbf{d}_i\$の絶対値の大きさが閾値以上のものを表面パーティクルとする。

しかしながら、この手法は、表面付近にないパーティクルであっても、SPHのように有効近傍範囲内のパーティクル密度に偏りがあると、表面パーティクルであると判定される(図3)。

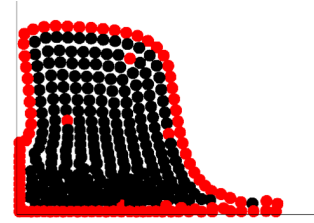


図 3: Solenthalerら[11]の方法による表面パーティクルの抽出

そこで、法線方向の近傍領域におけるパーティクル濃度値を用いて、そのようなパーティクルを除外する。これにより精度の高い表面パーティクルの抽出が可能となる。

表面パーティクルの法線方向における近傍領域のパーティクル濃度値\$c_i\$を下記の式(10)より求める。

$$c_i = \sum_j W_{poly6}((\mathbf{r}_i + \alpha \mathbf{n}_i) - \mathbf{r}_j) m_j \quad (10)$$

ここで、\$\mathbf{n}_i\$はそのパーティクルの法線、\$\alpha\$は法線を調整するための定数である。また\$W_{poly6}(\mathbf{r})\$は、Poly6カーネル関数[2]である。

通常、表面パーティクルの法線方向の近傍領域にはパーティクルがないため、濃度値は低くなる。よって求められた濃度値が高い場合、表面でない可能性が高く、これを表面パーティクルから除外する。図4にこの方法により表面を抽出した結果を示す。赤色で描画したパーティクルが表面パーティクルである。密度の違いに影響されず表面のパーティクルが抽出できる。

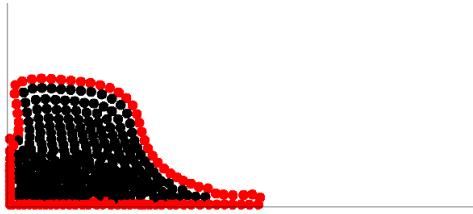


図 4: 改良された表面パーティクルの抽出

3.5 表面パーティクルのグループ化

複雑な形状を 1 つの関数で定義することは困難である。そこで表面パーティクルを小区間ごとのグループに分け、それぞれのグループごとに関数を定義し、グループ間の補間を行いながら、全体を統合することで、複雑な形状を表現する。

1 つのグループは中心座標、法線、領域半径の情報を持ち、領域内のパーティクルの法線方向がある程度同一となるようにする。そうすることで、図 5 のように法線方向が異なる領域が同一のグループになることはなく、領域内を単純な形状関数で定義することができる。また、全体を統合する際に補間方法を変えることで角のような鋭い形状特徴も失われない。

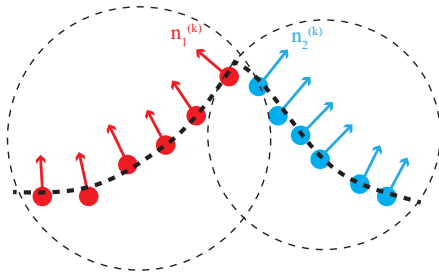


図 5: 法線情報によるグループ化

以下にグループ化の手順を示す。

- A1. グループ化されていない表面パーティクルの中からランダムに 1 つを選択し、そのパーティクルの位置、法線をそれぞれグループの中心点 \mathbf{x}_c 、法線 \mathbf{n}_c とする。
- A2. グループの法線 \mathbf{n}_c と近傍探索範囲 h_1 の i 番目の表面パーティクルの法線 \mathbf{n}_i から内積 $a_i = \mathbf{n}_c \cdot \mathbf{n}_i$ を求める。 $a_i \in [0, 1]$ が閾値 T_1 以上ならそのパーティクルをグループに含める。本論文では $T_1 = 0.96$ とした。
- A3. グループ内のパーティクルの平均位置、平均法線でそれぞれ \mathbf{x}_c 、 \mathbf{n}_c を置き換える。また、近傍探索範囲 h_1 に $2r_p$ を足して範囲を拡大する。 h_1 の初期値は $3r_p$ とした。ここで r_p はパーティクル半径である。

- A4. A2~A3 を新たにグループ化されるパーティクルがなくなるまで繰り返す。

新たにグループ化されるパーティクルがなくなると 1 つのグループ化が終了し、すべての表面パーティクルがいずれかのグループに属するまで A1~A4 を繰り返す。

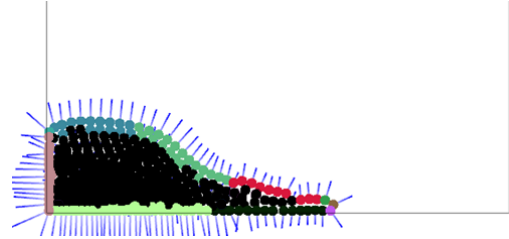


図 6: A1~A4 によるグループ化

このようにして、グループ化を行ったものを図 6 に示す。同色で示したパーティクルが同じグループに属するパーティクルである。図から法線方向に応じたグループ化ができてることがわかる。しかし、角や先端箇所などで属するパーティクルの数が少ないグループがある。フィッティングを正確に行うために必要な最少要素数 N_{min} は、グループ内に定義する形状関数の次数+1 となる。本論文では、フィッティングに用いる形状関数を 3 次多項式としているため、 N_{min} は 4 である。そのため、パーティクル数が N_{min} に満たないグループに以下の処理を加えることで、これを改善する。

- B1. グループ中心から近傍探索範囲 h_2 内にある表面パーティクルの法線の集合 $\mathbf{n} = \{\mathbf{n}_i | 0 \leq i \leq M\}$ のうち、グループの法線 \mathbf{n}_c に対して内積が最大となるもの ($a_{near} = \max(\mathbf{n}_c \cdot \mathbf{n}_i)$) を求める。なお h_2 は h_1 と同値の初期値をとる。
- B2. a_{near} が閾値 T_2 以上の場合、 a_{near} をもつパーティクルをグループに含める。この際、このパーティクルがすでに他のグループに属していても構わない。ただし、 $T_2 < T_1$ の値であり、本論文では $T_2 = 0.74$ とした。
- B3. グループ内のパーティクルの平均位置、平均法線でそれぞれグループの \mathbf{x}_c 、 \mathbf{n}_c を置き換える。また、近傍探索範囲 h_2 を h_1 と同様に拡大する。

B1~B3 をグループのパーティクル数が N_{min} 以上になるまで繰り返す。

この処理を行ってグループ化したものを図 7 に示す。図からすべてのグループのパーティクル数が、 N_{min} 以上となっていることがわかる。しかし、図 8 のように近傍のパーティクル数が少ない場合には、 N_{min} 以上パーティクル数を含んだグループ化はできない。これらのパーティク

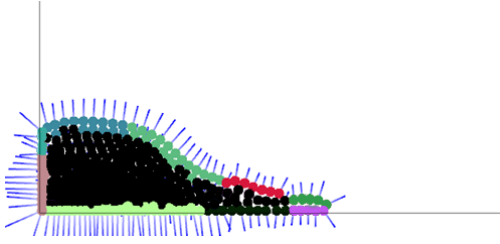


図 7: グループ化 (2)

ルを孤立パーティクルと呼ぶ。孤立パーティクルに対しては陰関数フィッティングが適用できないため、本論文では Müller ら [2] の方法によって陰関数場を生成する。

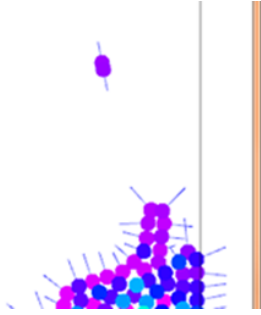


図 8: 孤立パーティクル

3.6 陰関数曲面フィッティング

Kanai ら [8] の方法を用いてフィッティングを行う。グループに属する N 個のパーティクルの中心座標 $\mathbf{P} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$ に対し、以下の式を満たすような形状関数を定義する。

$$f(\mathbf{P}) = y - (a_0 + a_1x + a_2x^2 + a_3x^3) = 0 \quad (11)$$

このとき、

$$\sum_{i=1}^N w_f(\mathbf{r}_i) f(\mathbf{r}_i)^2 \quad (12)$$

を最小化することで、未知の係数 $\mathbf{a} = \{a_0, a_1, a_2, a_3\}$ を決定する。但し、 w_f は重み関数である。

3.7 陰関数場の生成とレンダリング

各グループ領域内で定義した形状関数をグループ間の補間を行い、陰関数場を生成する。任意の点 \mathbf{x}_i における陰関数値 C 、および法線 $\tilde{\mathbf{n}}$ を次の式で求める。

$$C = \frac{\sum_j w_j c_j}{\sum_j w_j} \quad (13)$$

$$\tilde{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|}, \quad \mathbf{n} = \frac{\sum_j w_j \mathbf{n}_j}{\sum_j w_j} \quad (14)$$

ここで、 w_j は下式に示すグループ中心から \mathbf{x} までの距離 d_j による重み関数である。

$$w_i(d_j, b) = \frac{1}{\sqrt{2\pi b^2}} \exp\left(-\frac{d_j^2}{2b^2}\right) \quad (15)$$

なお、 b はグループ法線間の角度が大きくなるほど線形に減少する値である。グループ法線間の角度が大きくなるほど、 w_j は急な変化になる。これにより、グループ法線間の角度が小さい平らな箇所では緩やかな面を、角度が大きい先端箇所では鋭いエッジを表現できる。

また、グループ化の際に、近傍に必要なパーティクル数がない場合、Müller ら [2] の方法によって陰関数場を生成し、正規化して陰関数場に統合する。

4 結果

提案手法を実装し、既存手法とともにその結果を示す。実験は、Intel Core i5-2400 3.10GHz, NVIDIA GeForce GTX 580 の環境で行った。提案手法の描画結果と計算時間を比較するために、流体力学において一般的に用いられる水柱の崩壊シミュレーション (Dam breaking) と水柱を両端に配置したシーン (Double dam breaking) の 2 つのシーンで実験を行った。

シーン 1: Dam breaking 水槽の片側から水柱が崩れる様子をシミュレーションした。なお、総パーティクル数 N_p は 325 に設定した。既存手法の描画結果を図 9 に、提案手法の描画結果を図 10 に示す。それぞれの方法で抽出された表面を赤い線で示した。Dam breaking のシーンでは水柱が重力で下方向に押しつぶされるため、SPH 法では粒子密度が上部と下部で大きく異なってしまうが、我々の方法ではそのような状況でも、正しく表面パーティクルを抽出し、フィッティングすることに成功している。

シーン 2: Double dam breaking 水槽の両端から水柱が崩れる様子をシミュレーションした。なお、総パーティクル数 N_p は 650 に設定した。既存手法の描画結果を図 11 に、提案手法の描画結果を図 12 に示す。

壁に水が衝突する場面や中央で水が衝突する場面で、既存手法は波の先端が丸くなっているが、提案手法では、波の先端の鋭さが表現できていることがわかる。これは、法線によるグループ化やグループ法線間の角度による合成が適切に働いているためである。しかし、アニメーションにすると、ステップごとにフィッティングの範囲が異なるため、表面が振動するという現象が確認された。これを解決するために、前ステップのグループ化の結果やフィッティ

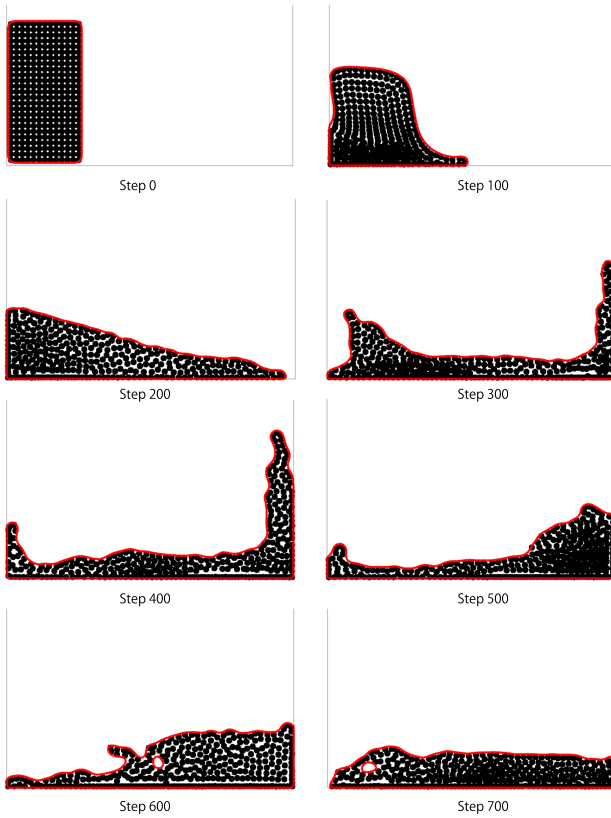


図 9: シーン 1 を既存手法で描画した結果

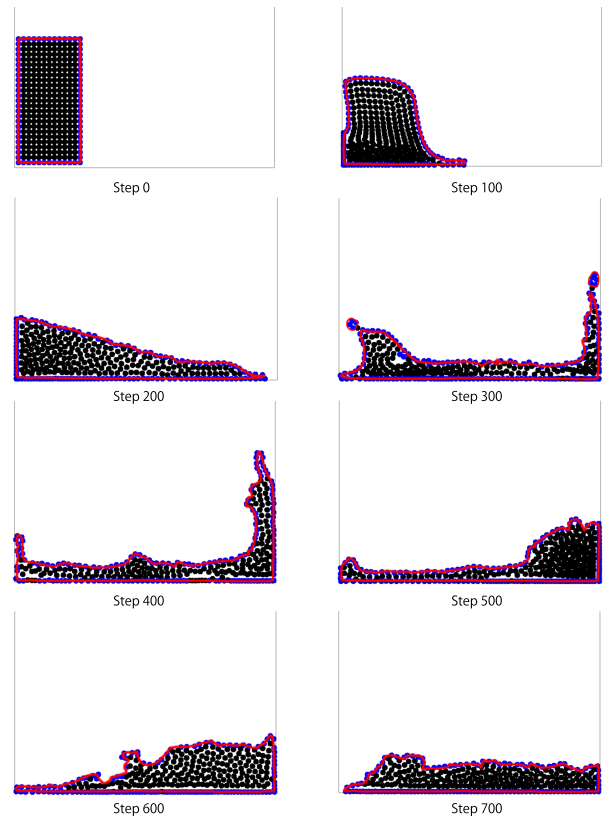


図 10: シーン 1 を提案手法で描画した結果

ングの情報を次のステップに反映させ、フレーム間の動きを滑らかにする必要がある。

表 1: 計算時間 (fps)

手法	既存手法	提案手法
Dam breaking	40.0	3.9
Double dam breaking	32.5	1.5

それぞれの実験の 700step までの平均計算時間を表 1 に示す。既存手法に比べ、シーン 1 において 10 倍、シーン 2 において 22 倍となった。これは、グループ化の際の内積計算とフィッティングにかかる計算コストが高いためだと考えられる。しかし、これらの処理は並列計算が容易であり、GPU による計算を用いることで、大幅に計算時間を短縮できると考えられる。

5 結論と今後の研究

本論文では、SPH における陰関数曲面フィッティングを用いた液体表面可視化手法を提案した。ポイントベースモデリングのように数百万から数億もの大規模な点群に対して使われる陰関数フィッティングをリアルタイムのシミュレーションで使われる数百から数万程度のパーティク

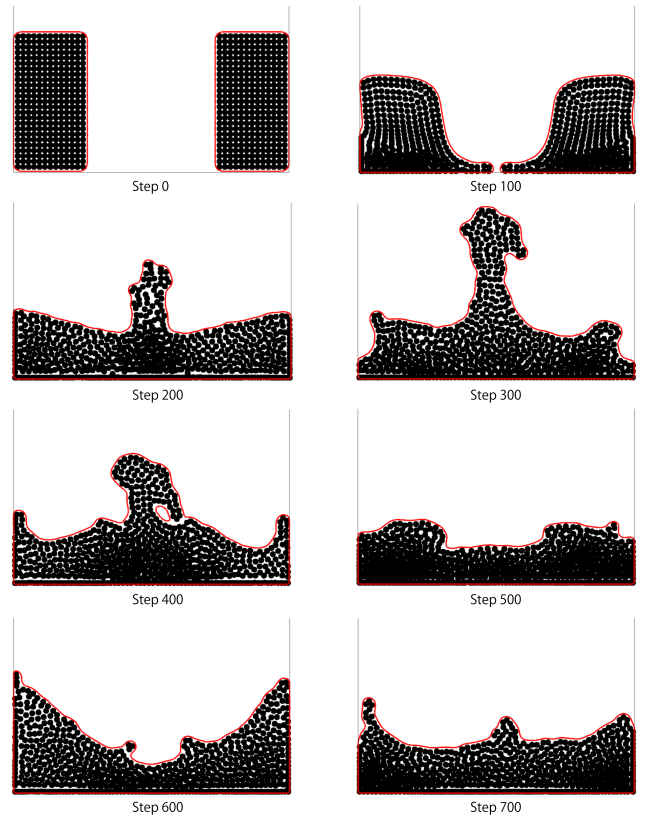


図 11: シーン 2 を既存手法で描画した結果

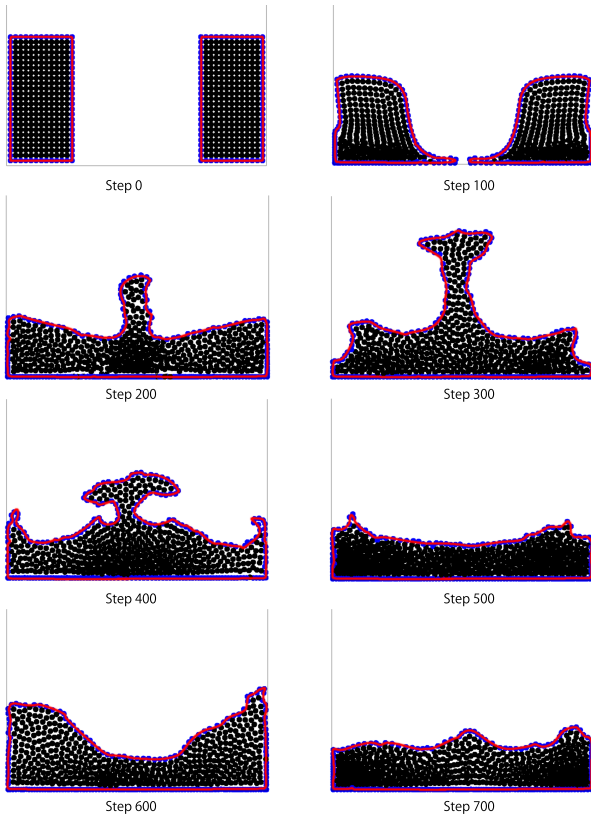


図 12: シーン 2 を提案手法で描画した結果

ルに適用するためのグループ分けや合成の手法を開発し、液体表面の滑らかかつ、波などの鋭い形状の表現を可能とした。

実験では、法線によるグループ化、陰関数の合成方法の工夫により、既存の方法に比べ、波の衝突などにより発生する鋭い形状を再現することができた。ただし、フィッティングの結果がシーンによって微妙に異なり、アニメーションにすると表面が振動して見えてしまった。これは、前ステップのグループ化の結果やフィッティングの情報を次ステップに影響させることで、解決できると考えられる。

今後の研究としては、3次元シミュレーションへの適応があげられる。本論文で提案した手法はほとんどの場合、パーティクルの法線情報を用いる。特に法線間の角度の算出などは3次元化が容易であり、提案手法は3次元シミュレーションにも適用可能と考えられる。また提案手法は計算コストが高く、計算時間がかかっている。GPUを用いた並列計算により、グループ化の際の内積計算やフィッティングで高速化することでリアルタイムでのレンダリングを実現する。今回、陰関数フィッティングの形状関数に3次の多項式を用いたが、ほかの関数でも実験を行い適切な関数を検討したい。さらに、グループ化の際にパーティクル数が少ない場合、既存手法を用いたが、これもフィッティングに用いる関数を変えることで対応したい。加えて、パーティクル数を増やした場合に適切に動作するかの

確認、しぶきの表現、クラスタリングによるグループ化なども検討していく。

参考文献

- [1] J. Blinn, “A generalization of algebraic surface drawing”, *ACM Transactions of Graphics*, 1(3), pp.235-256, 1982.
- [2] M. Müller, D. Charypar and M. Gross, “Particle-based fluid simulation for interactive applications”, In *Proceeding of the ACM SIGGRAPH Symposium on Computer Animation*, pp.154-159, 2003.
- [3] J. Yu and G. Turk, “Reconstructing surface of particle-based fluids using anisotropic kernels”, In *Proceeding of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.217-225, 2010.
- [4] B. Adams, M. Pauly, R. Keiser and J. L. Guibas, “Adaptively sampled particle fluid”, *ACM Transactions of Graphics (Proc. SIGGRAPH2007)*, 26(3), 48, 2007.
- [5] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin and C. T. Silva, “Computing and rendering point set surfaces”, *IEEE Transactions on Visualization and Computer Graphics*, 9(1), pp.3-15, 2003.
- [6] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum and T. Evans, “Reconstruction and representation of 3D objects with radial basis functions”, In *Proceeding of SIGGRAPH 2001*, pp.67-76, 2001.
- [7] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk and H.-P. Seidel, “Multi-level partition of unity implicits”, *ACM Transactions on Graphics (Proc. SIGGRAPH2003)*, 22(3), pp. 463-470, 2003.
- [8] T. Kanai, Y. Ohtake, H. Kawata and K. Kase, “GPU-based rendering of sparse low-degree implicit surfaces”, In *GRAPHITE '06: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, pp.165-171, 2006.
- [9] M. Becker and M. Teschner, “Weakly compressible SPH for free surface flows”, In *Proceeding of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.209-217, 2007.

- [10] B.Solenthaler and R. Pajarola, "Predictive-corrective incompressible SPH", In Proceeding of SIGGRAPH 2009, pp.1-6, 2009.
- [11] B. Solenthaler, Y. Zhang and R.Pajarola, "Efficient refinement of dynamic point data", Eurographics Symposium on Point-Based Graphics, pp.65-72, 2007.