

粒子ベース流体シミュレーションデータの圧縮

Compression of Particle-based Fluid Simulation Data

和田 耕太† 岩崎 慶†
Kota Wada Kei Iwasaki

1. はじめに

粒子法による流体シミュレーションは、3DCG による映像制作において重要な手法の一つであり、流体力学に基づいたリアルな映像を作成できるのが魅力である。しかしながら、そのシミュレーションデータはすべての粒子について、全フレームでの座標を保持する必要があり、非常にデータサイズが大きくなるという問題がある。

そこで、本研究ではシミュレーションに含まれる粒子を小さなクラスタに分け、クラスタごとに主成分分析を用いて粒子の位置データを圧縮する手法を提案する。本研究により、見た目に遜色がなくシミュレーションデータを 95% 圧縮することができる。

2. 関連研究

関連研究としては、3D アニメーションデータの圧縮の研究が挙げられる。Alexa ら[1]は、主成分分析を用いて 3D アニメーションの各キーフレームのメッシュデータを圧縮する手法を提案した。この手法を拡張した手法[2,3,4]や Iterative Closest Point 法[5]を用いた手法が提案されているが、これらは全てメッシュによる 3D 形状およびアニメーションを対象としたものであり、粒子法データの圧縮手法は提案されていない。

3. 提案手法

本研究では、粒子法シミュレーション手法として Smoothed Particle Hydrodynamics 法を用いるが、他の粒子ベース流体シミュレーション手法にも適用可能である。

3.1 概要

粒子法のシミュレーションで得られるデータは、すべての粒子について、すべてのフレームにおける座標を羅列したものである。これを並べ、行数が粒子数×3 (次元)、列数がフレーム数の行列 X として取り扱う。このようにデータを行列として表現すれば、これを特異値分解によって分解することができる。

特異値分解によって得られた行列は、もとの行列を近似するにあたって必要な成分をその重要度順に含んでいる。これを利用し、重要でない成分を切り捨てることによってデータを非可逆的に圧縮する。

3.2 クラスタリング

流体シミュレーションに用いられる粒子は通常数万～数十万粒子であるため、全粒子について 1 つの行列として表現して特異値分解することは難しい。そのため、シーンに

含まれる粒子を小さなクラスタに分け、クラスタごとに特異値分解を行う。

クラスタリングには K-means 法を用い、圧縮するフレームの内最初のフレームの座標をもとにクラスタリングする。また、クラスタ 1 つあたりの粒子数を予め決めておき、これを C とする。また、K-means 法でクラスタリングをするにあたり、目標のクラスタ数 N_c が必要になる。これは全粒子数 N_p を C で割れば求められるが、この方法で計算した N_c を利用してそのまま K-means 法でクラスタリングするとクラスタの粒子数が極端に多くなったり少なくなったりしてしまう。これは後に特異値分解する際の妨げになるので、本手法ではまずこの N_c を 1/10 倍にしておき、おおまかにクラスタリングしてからそのクラスタ内の粒子を C 個ずつに分割する。これによりクラスタごとの粒子数のばらつきが少なくなり、クラスタリングの高速化も期待できる。この方法では K-means 法により得られる局所性は 1/10 になってしまうが、クラスタの大きさは全粒子数に対して十分に小さいので問題は起きない。

以下に具体的なアルゴリズムを示す。

1. 粒子数 N_p を 1 クラスタあたりの大きさ C で割り、その 1/10 をクラスタ数 N_c とする
2. すべての粒子について、ランダムにクラスタを割り当てる
3. 各クラスタについて、その座標の平均を求める
4. 各粒子について、全てのクラスタの平均座標と比較し最も近いクラスタに割り当て直す
5. すべての粒子についてクラスタの割り当てが変化しなければ 6 に進み、そうでなければ 3 から繰り返す
6. 割り振ったクラスタを、目的のクラスタの大きさ C に近くなるように分割する

3.3 主成分分析

圧縮前の粒子法によるシミュレーションデータは各粒子の 3 次元座標を対象のフレーム分並べたものである。これを(1)式のように並べ、クラスタごとに 3C 行 F 列 (C はクラスタ内粒子数、 F はフレーム数)の行列 X として扱う。 i 番目の粒子の j フレーム目における座標を p_{ij} とすれば、行列 X は以下の式(1)で計算される。

$$X = \begin{pmatrix} p_{11} & \cdots & p_{1F} \\ \vdots & \ddots & \vdots \\ p_{3C1} & \cdots & p_{3CF} \end{pmatrix} \quad (1)$$

主成分分析を用いる前に、まずクラスタに属する全ての粒

子それぞれについて、全フレームにおける座標の平均値を求め、その値を全フレームの座標から引き、これを X' とする。 X' を特異値分解すると以下のように 3 つの行列に分解することができる。

$$X' = USV^T \quad (2)$$

(2)式で行列 U は $3C$ 行 $3C$ 列の直交行列、行列 S は $3C$ 行 F 列で対角成分以外は 0、行列 V^T は F 行 F 列の直交行列(転置行列)である。行列 S の対角成分は特異値であり、各成分の寄与率を表している。特異値は 1 行 1 列から順に単調減少する。つまり、この分解を行った時点で寄与率の高い順に成分が並んでいる。

特異値分解した行列をもとに、 X' を以下のように近似することができる。

$$\tilde{X} = U\tilde{S}V^T \quad (3)$$

(3)式の \tilde{S} はその対角成分(特異値)のうち大きい方から特定の数だけ残して他を 0 としたものである。この残す特異値の数を要素数 R とする。この R の大小により圧縮率が決まる。

ここで、行列 U のうち R 列、行列 S のうち R 個の特異値、行列 V のうち R 行を使うので、これ以外の情報は保持しなくてよい。これらを除いた行列のデータと、最初に求めた平均値のデータを出力すれば圧縮は完了となる。

圧縮したデータから i 番目の粒子の k フレーム目の座標データを取り出すには、行列 U の i 行目の要素のうち R 個、行列 \tilde{S} の特異値のうち R 個、行列 V の k 列目の要素のうち R 個をそれぞれ順に掛けあわせ、最初に求めた平均位置にすべて足しあわせればよい。

4. 結果と評価

本研究の評価は、圧縮前後の結果画像の比較、粒子位置の比較、及びデータの圧縮率をもとに行う。なお、実験に使用した流体シミュレーションは 240,000 粒子、900 フレームのものであり、座標は全て float 型で保持されている。評価に用いているのはそのうち 601~900 フレームについて、1 クラスタあたり 500 粒子として圧縮したものである。1 クラスタあたりの粒子数 C や使用する特異値の数 R の値を変化させたときの結果の違いについては後述する。

4.1 圧縮率の計算法

圧縮前のデータサイズ A は、1 クラスタあたりの粒子数 $C \times 3$ 次元 \times フレーム数 $F \times 4$ バイトなので、

$$A = 12CF$$

で求められる。今回の例では $C=500$ 、 $F=300$ なので、 $A=1,800,000$ (バイト) である。圧縮後の大きさは、使用する特異値の数 R に依存する。元のデータを特異値分解した 3 つの行列のうち行列 U からは、クラスタ内粒子数 C 行それぞれから R 個の要素が必要なので、 $C \times 3$ 次元 $\times R \times 4$ バイト必要である。行列 \tilde{S} からは、特異値を必要な数だけ利用するので、 $R \times 4$ バイト、行列 V^T からはフレーム数 F 行それぞれから R 個の要素が必要なので、 $F \times R \times 4$ バイト必要となる。また、圧縮したデータから座標を取り出す際に各粒子(C 個)について全フレームにおける 3 次元の平均座標が必要になるので、これに $C \times 3 \times 4$ バイト必要となる。以上をまとめると、圧縮後のデータサイズ B は以下の式で求められる。

$$B = 12CR + 4R + 4FR + 12C \quad (4)$$

ここで、 A に対する B の比率は B/A で求められ、 A に対して削減できた割合を百分率で表したものを圧縮率 E とすると、 E は

$$E = \left[1 - \frac{\{(3C+1+F)R+3C\}}{3CF} \right] * 100 \quad (5)$$

で求めることができる。

4.2 圧縮率と誤差の評価

前節に基づいて圧縮率を計算すると、表 1 のようになる。

表 1 使用する特異値の数 R と圧縮率

特異値の数 R	圧縮後サイズ	圧縮率
24	178,896	90.06%
12	92,448	94.86%
3	34,816	98.06%
2	20,408	98.86%

このように、打ち切る特異値の数 R が 24 で約 90%、12 で約 95% という圧縮率を達成している。特異値分解した際に得られる特異値は、各要素の寄与率を表しており、これをグラフにプロットするとどの程度の要素数が必要かを知ることができる。特異値の数と寄与率のグラフを図 1 に示す。

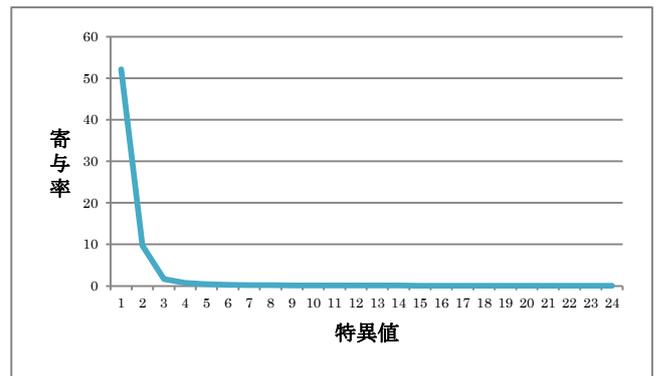
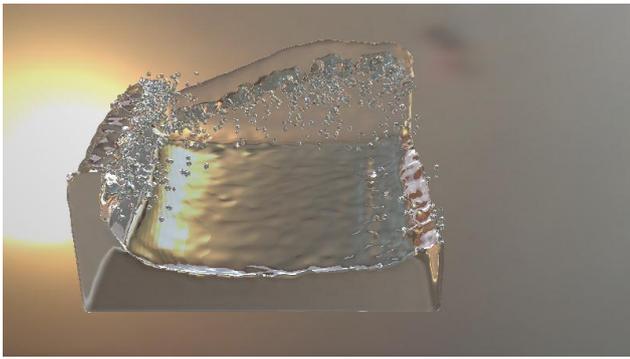


図 1 各特異値とその寄与率のグラフ

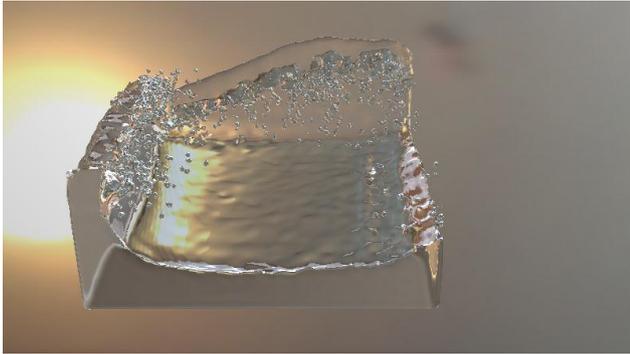
図 1 より、要素数 3 以降はほとんどもとの成分を表すのに寄与していないことがわかる。要素数を 3 にした場合、圧縮率は 98.6% を超える。

圧縮の際の誤差に関しては、出力画像の比較と、全粒子について圧縮前後の座標のユークリッド距離の累積によって評価する。

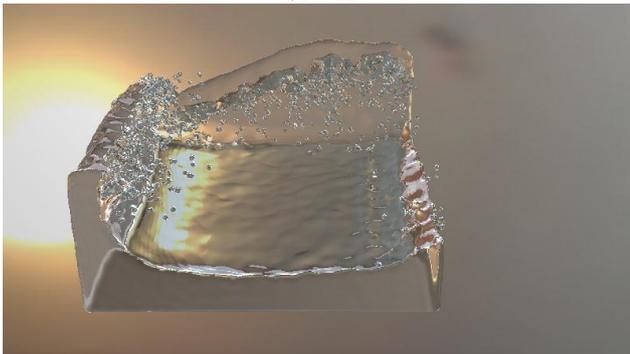
要素数を変えた場合の圧縮結果画像を図 2 に示す。図 2(a) と (b) を比較して、要素数 12 (圧縮率約 94.86%) でもほとんど誤差は見られない。ドット単位で水面のゆらぎの位置が違う程度である。要素数 3 (圧縮率約 98.06%) では水しぶきの位置などが目に見えて違う場合があるが、水面自体には目立った誤差を視認できない。



a)非圧縮



b) $R=12$

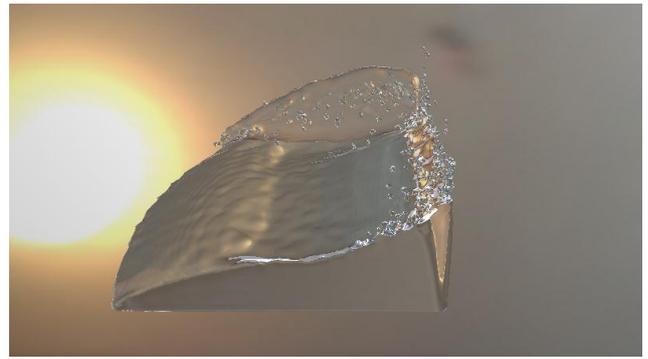


c) $R=3$

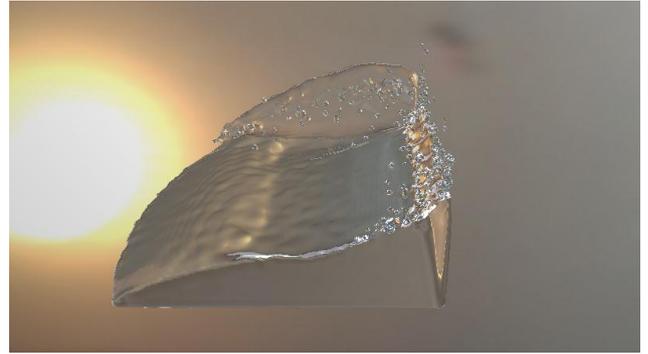
図2 圧縮結果の例。a)非圧縮のデータをもとにレンダリングした結果 b) $R=12$ に圧縮した結果(圧縮率 94.86%) c) $R=3$ に圧縮した結果(圧縮率 98.06%)

また、同一シミュレーション中でもフレームによって結果が大きく違うことがある。図3のシーンでは誤差が顕著に現れている。

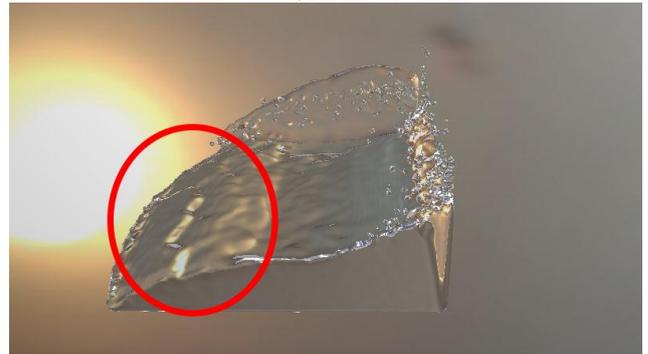
図3より、要素数12では結果画像からほとんど誤差は見取れないが、要素数3の結果画像では水面の様相が大きく崩れている。このように大きな誤差が発生する場合もあるが、その際は要素数を調整することで用途に合わせて圧縮率と結果画像の誤差のバランスを取ることができる。



a)非圧縮



b) $R=12$



c) $R=3$

図3 圧縮結果のうち誤差が顕著に出た例。a)非圧縮のデータ b) $R=12$ に圧縮した結果 c) $R=3$ に圧縮した結果(赤丸の部分が他の結果と大きく違っている)

4.3 誤差の定量評価

前節までは結果画像をもとに誤差を評価したが、本節では圧縮前後の粒子の座標の差により圧縮結果を定量的に評価する。

誤差を求めるには、全粒子について圧縮後の座標と元の座標のユークリッド距離を求め、その総和を求めれば良い。ただし、これだけではシミュレーション全体の座標の大きさに依存して誤差が大きく算出されてしまうので、各粒子の座標の絶対値の総和で割る。 i 番目の粒子の x, y, z 座標をそれぞれ p_{xi}, p_{yi}, p_{zi} 、圧縮後の i 番目の粒子の x, y, z 座標をそれぞれ $\tilde{p}_{xi}, \tilde{p}_{yi}, \tilde{p}_{zi}$ として、累積誤差 D を

$$D = \frac{\sum_{i=1}^C \sqrt{(p_{xi} - \tilde{p}_{xi})^2 + (p_{yi} - \tilde{p}_{yi})^2 + (p_{zi} - \tilde{p}_{zi})^2}}{\sum_{i=1}^C \sqrt{p_{xi}^2 + p_{yi}^2 + p_{zi}^2}} \quad (6)$$

と定義する。圧縮時に打ち切る特異値の数 R を変化させた

時の累積誤差のグラフを図4に示す。

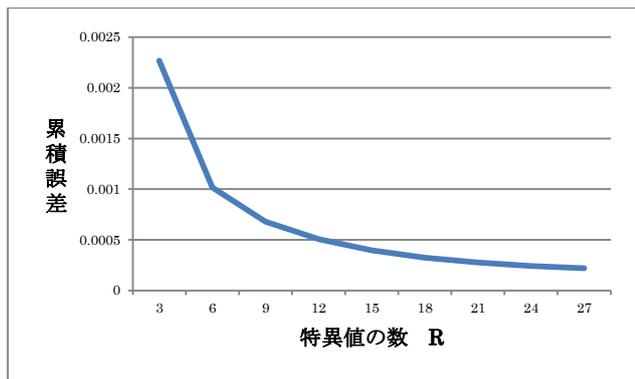


図4 使用する特異値の数 R と累積誤差の関係

図4により、 $R=9$ 以降はさほど結果が改善しないことがわかる。 $R=9$ で打ち切れば、(5)式より、圧縮率は96.06%である。

4.4 特異値の数以外の値による圧縮結果への影響

先程までの例では $C=500$ としていた。これを小さく変更すると累積誤差が増加し、クラスタリングにかかる時間も増加してしまう。逆に C を大きくすると誤差は減少する。 R を調整して圧縮率を95%に近く保ち、 C を変化させて誤差累計を調べた結果を図5に示す。

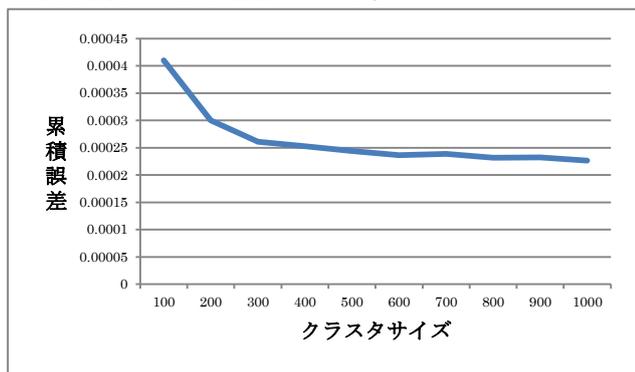


図5 クラスタサイズと累積誤差の関係

図5より、クラスタサイズが大きくなると誤差が小さくなるのがわかる。したがって、基本的にはコンピュータの性能が許す限りクラスタサイズは大きくすればよい。ただし、大きくしすぎれば K-means 法によるクラスタの局所性が失われ、誤差が増大する可能性がある。なお、この実験の誤差は $C=700$ と $C=900$ のときに単調減少ではなくなっているが、これは圧縮率を約95%に調整する際に使用する特異値の数 R を整数に丸めたことに起因する誤差である。

また、今回の実験ではクラスタリングに K-means 法を使用した。K-means 法を使用せず、シミュレーションデータの順に C ずつクラスタリングする方法に比べて誤差は2.90~2.70%減少した。

さらに、シミュレーション自体のタイムステップが大きいと、少ないフレームでも粒子が複雑な動きをするので良好な結果を得るにはより多い特異値の数が必要になると考えられる。

5. まとめと今後の課題

本研究では粒子ベース流体シミュレーションデータを、クラスタ主成分分析を用いて圧縮し、高い精度を保ちつつ95%以上のデータを削減することに成功した。

今後の課題としては、粒子が途中で消失するデータに対して圧縮を行うことが挙げられる。提案手法は特異値分解を用いるため、同じ粒子番号に対しては全フレームにわたって同じ粒子が対応していなければならない。粒子が消失した際にその粒子の座標を保持しないデータ、つまりフレームごとに粒子の総数が違う場合があるデータを圧縮しようとすると、別の粒子の座標ごと特異値分解することになり、うまく圧縮できない。そのため、フレーム間で各粒子の動きを予測し、対応していそうな粒子を探しうまく特異値分解できるようにして解決することが考えられる。

参考文献

[1] Marc Alexa and Wolfgang Müller, "Representing Animations by Principal Components", Computer Graphics Forum, Vol. 9, No. 3, pp.411-418, 2000.

[2] Zachy Karni and Craig Gotsman, "Compression of Soft-body Animation Sequences", Computers & Graphics, Vol. 28, No. 1, pp. 25-34, 2004.

[3] Rachida Amjoun, Ralf Sondershaus, and Wolfgang Straßer, "Compression of Complex Animated Meshes", Proc. of Computer Graphics International, pp.606-613, 2006.

[4] Mirko Sattler, Ralf Sarlette, and Reinhard Klein, "Simple and Efficient Compression of Animation Sequences", Proc. of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.209-217, 2005.

[5] Sumit Gupta, Kuntal Sengupta, and Ashraf A. Kassim, "Compression of Dynamic 3D Geometry Data Using Iterative Closest Point Algorithm", Computer Vision and Image Understanding, Vol.87, No. 1-3, pp. 116-130, 2002.