Vol.2022-CG-185 No.5
2022/3/11

2D乱流合成を用いた煙シミュレーションの高速化

笹　晃輔[1,a)]　土橋　宜典[1,b)]　佐藤　周平[2,c)]

概要：コンピュータグラフィックスにおいて流体の表現は映画や TV ゲーム等，様々なアプリケーションで利用されている．シミュレーションやレンダリング技術の向上により非常に写実的な表現が可能となっているが，その計算には高いコストを要する．また，ユーザの望む映像を作成するためには，パラメータを調整する煩雑な作業が必要となる．これを解決する手段として，後処理的に乱流を転写する方法が提案されている．この方法では低解像度の流れに対し，別の高解像度の流れからその乱流成分のみを転写することで，所望の流れを作成する時間や手間を削減できる．しかし，この方法には計算コストの高い処理が含まれており，大幅な計算の高速化には至っていない．そこで我々は，2 次元で乱流転写を実行した流れをもとに 3 次元の流れを再構築することで従来手法の高速化を図る．提案手法により従来法と同等の視覚的クオリティを保ちながら，計算を約 12 倍高速化することに成功した．

キーワード：流体シミュレーション，乱流合成，スタイル転写

Efficient Smoke Simulation Using 2D Turbulent Style Transfer

Abstract: In computer graphics, fluid simulation is used in various applications such as movies and video games. Due to the advancement of simulation and rendering techniques, realistic animations can be created, but some problems still remain: the simulation and the rendering require huge computation cost, and tedious parameter adjustment is needed to create the desired results. To address this problem, a post processing method for transferring turbulence has been proposed. This method can reduce the computational cost for creating the desired fluid flows by transferring the turbulent motion from a high-resolution flow to another low-resolution flow. However, this method contains computationally expensive processes, so its computation time is still long. We attempt to speed up this previous method by reconstructing the 3D flow based on the 2D flow with transferred turbulence. Our method achieves approximately 12 times faster computation than the previous method while maintaining the same visual quality as the previous method.

Keywords: fluid simulation, turbulence synthesis, style transfer

1. まえがき

近年，コンピュータグラフィックスによる流体の表現は映画やテレビゲーム，市販ツールなどの様々な産業分野で利用されている．その背景には，流体の動きを記述するナビエ・ストークス方程式を解くことにより流体の挙動を可視化する，流体シミュレーションという技術があるが，これには膨大な手間と時間を要する．これを解決する手段の 1 つとして，低解像度のシミュレーションにより大域的な流れを作成し，後処理として乱流を補完して高解像度化を行う，乱流合成と呼ばれる手法がある．佐藤らは 2018 年に，テクスチャ合成を応用した例示ベースの乱流合成手法（以下，乱流転写とする）を提案し，簡単かつ直観的な乱流合成を可能にした [1]．しかし，そのアルゴリズムには計算コストの面で更なる改善の余地が残されている．

2 次元の高解像度シミュレーションを活用して 3 次元の流れを高速生成する，Rasmussen らの手法がある [2]．本研究ではこの手法から着想を得て，佐藤らの乱流転写をより高速に実行するための手法を提案する．流体の対象を煙に絞り，2 次元の乱流転写を用いて 3 次元の高解像度の煙を効率的に生成することを目的とする．

[1] 北海道大学
Hokkaido University
[2] 富山大学
Toyama University
a) sasa@ime.ist.hokudai.ac.jp
b) doba@ime.ist.hokudai.ac.jp
c) ssato@eng.u-toyama.ac.jp

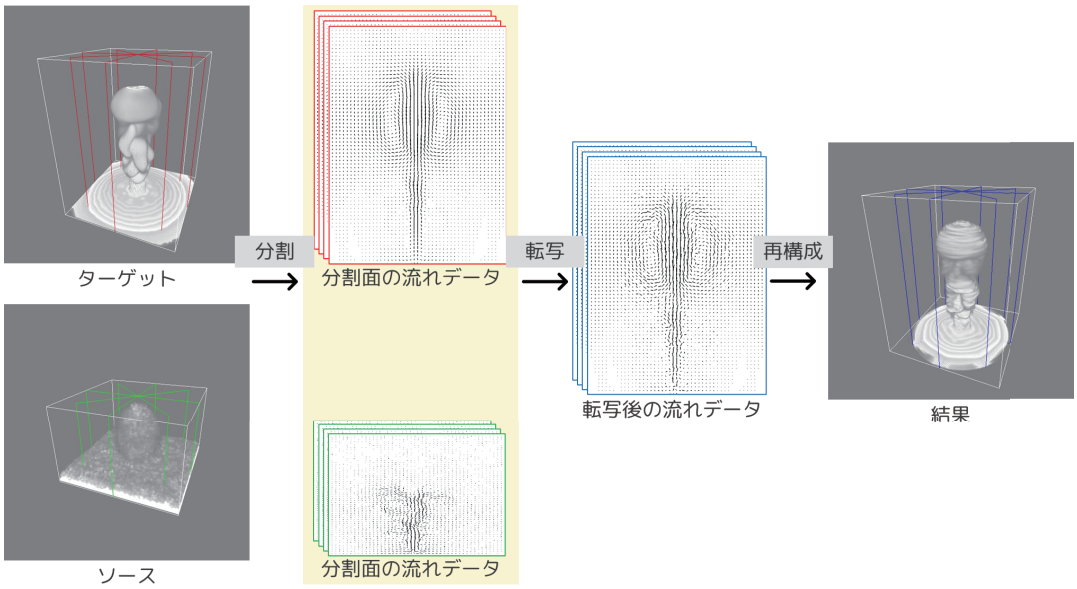

図 1: 提案手法の処理の流れ

2. 関連研究

CG 流体シミュレーションの基礎が確立されてから，乱流合成による高速化に関する研究が数多く行われてきた．Theodore Kim らは 2008 年に，ウェーブレットノイズを用いて低解像度の速度場に対して高周波成分を補完する手法を提案した [3]．Tobias Pfaff らは 2009 年に，物体に対して人口境界層という概念を導入し，流体と物体との相互作用により発生する乱流を再現する手法を提案した [4]．Michael B. Nielsen らは同年に，所望の低解像度の流れに従うよう高解像度の流れをシミュレーションする手法を提案した [5]．佐藤らは 2018 年に，画像処理分野のテクスチャ合成と呼ばれる技術を応用した，例示ベースの乱流合成手法を提案した [1]．

乱流合成とは別の分野で，2D 流体を活用して 3D 流体シミュレーションを高速化する研究もいくつか行われてきた．Nick Rasmussen らは 2003 年に，2D 流体シミュレーションを活用して大規模な煙シミュレーションを高速に実行できる手法を提案した [2]．

本研究の基盤となる佐藤らの乱流転写手法 [1] と Rasmussen らの手法 [2] を順に紹介する．佐藤らの手法 [1] では，低解像度の煙（ターゲット）と高解像度の煙（ソース）を入力とする．ターゲットは，所望の煙の大域的な流れを設計したものとなっており，ソースは，ターゲットに補完したい乱流成分を多分に含む流れとなっている．映像制作現場では，流体映像の製作過程で数多くのデータが作られる．ソースの入力としてそういったデータを再利用することで，手間と時間を節約することができる．この手法は，ターゲットの空間をパッチと呼ばれる立方体に分割し，パッチ単位で実行される．煙の存在するパッチごとに，ソースから流れの類似する領域を探索し，その高周波成分を転写する．そして，パッチ間に不連続な流れが生じる場合を考慮して，乱流転写を行ったパッチの境界に対して流れの平滑化を適用する．この手法により簡単かつ直観的に乱流合成を行えるようになったが，そのアルゴリズムには計算コストの高い処理が含まれており，大幅なシミュレーションの高速化には至っていない．

Rasmussen らの手法 [2] では，2 次元の高解像度シミュレーションを数回実行した後に，それらを 3 次元空間における流れの断面図として用いる．そして，これらの断面の間の空白の領域を線形補間により埋める．このままでは回転対称で非現実的な煙となってしまうため，最後にコルモゴロフ則に従う周波数スペクトルを利用して流れにランダムな動きを補完する．この手法により，大規模な核爆発のような現象により発生する煙のシミュレーションを非常に高速に実行できる．

3. 提案手法

本節では 2 次元の乱流転写を用いて高解像度の煙を効率的に生成する手法を提案する．提案手法の処理の流れを図 1 に示す．入力は 3 次元格子で表現された速度場とし，佐藤らの手法と同様に低解像度の速度場（ターゲット）と高解像度の速度場（ソース）の 2 種類を用いる．これらの速度場から複数の断面を取得し乱流転写処理を実行することで，乱流が付加された高解像度の断面を取得する．ここで，断面における速度場も格子で表現するものとする．そして，高解像度の断面を基に 3 次元の流れを再構築することで，高解像度の煙を生成する．

3.1 断面取得処理

本節では，提案手法の処理の第 1 段階である分割処理について説明する．分割処理では，煙の中心軸と外縁全体を含むような形で放射状の断面を考え，断面での速度分布を取得した後に 2 次元の乱流転写を適用する．図 2 にその模式図を示す．断面の数を N_{slice} とし，これはユーザ指定に

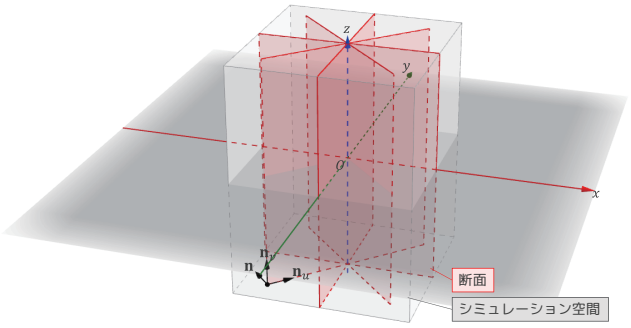

図 2: 分割処理の模式図

より与えられるものとする．$i(=0,1,2,\cdots,N_{slice}-1)$ 番目の断面を S_i とおくと，S_i は以下の方程式 (1) で表される平面となる．

$$\boldsymbol{n}_i \cdot \boldsymbol{p} = 0 \tag{1}$$

ここで，$\boldsymbol{p}$ は断面上の任意の一点，$\boldsymbol{n}_i$ は S_i の法線ベクトルであり，S_i と xz 平面とのなす角を θ_i としたとき，$\boldsymbol{n}_i = (-\sin\theta_i, \cos\theta_i, 0)$ と与えられる．$\boldsymbol{p}$ における流れの速度を $\boldsymbol{v}_{3D}$ とおくと，これは $\boldsymbol{p}$ の周囲の流れの速度から線形補間により求められる．ただし，補間はシミュレーション空間の x 成分と y 成分のみに対して行われる．線形補間の模式図を図 3 に示す．$\boldsymbol{p} = (x, y, z)$ とおく．この

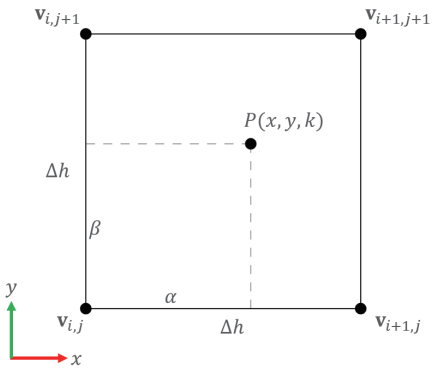

図 3: 分割処理における線形補間

とき，ターゲットの格子幅を Δh とすると，周囲の格子番号は $i = \mathrm{floor}(\frac{x}{\Delta h})$，$j = \mathrm{floor}(\frac{y}{\Delta h})$ で求められる．また，$\alpha = \frac{x - i\Delta h}{\Delta h}$，$\beta = \frac{y - j\Delta h}{\Delta h}$ とすると，$\boldsymbol{p}$ における速度 $\boldsymbol{v}_{3D}$ は以下の補間式 (2) により求められる．

$$\begin{aligned}\boldsymbol{v}_{3D} = {} & \boldsymbol{v}_{i,j}(1-\alpha)(1-\beta) + \boldsymbol{v}_{i+1,j}\alpha(1-\beta) \\ & + \boldsymbol{v}_{i,j+1}(1-\alpha)\beta + \boldsymbol{v}_{i+1,j+1}\alpha\beta\end{aligned} \tag{2}$$

最後に，以下の式 (3) によって断面に投影された 2 次元の速度 $\boldsymbol{v}_{2D} = (u, v)$ に変換する．

$$u = \boldsymbol{v}_{3D} \cdot \boldsymbol{n}_u, v = \boldsymbol{v}_{3D} \cdot \boldsymbol{n}_v \tag{3}$$

ここで，$\boldsymbol{n}_u$，$\boldsymbol{n}_v$ はそれぞれ S_i における u 方向，v 方向の単位ベクトルであり，$\boldsymbol{n}_u = (\cos\theta_i, \sin\theta_i, 0)$，$\boldsymbol{n}_v = (0, 0, 1)$ と表される．以上の分割処理をターゲットとソースの両方に対して適用し，それぞれの断面の速度場を取得する．

3.2 乱流転写処理

本節では，提案手法の第 2 段階である乱流転写処理について説明する．提案手法では，3 次元の流れを断面に分割し，次元を 1 つ下げた状態で乱流転写を実行する．2 節で述べた通り，乱流転写処理はパッチごとの乱流転写と，パッチ間境界の平滑化の 2 段階の処理を踏んで実行される．佐藤らの従来手法では，3 次元の乱流転写であるためパッチが立方体領域となり，そのパッチ探索処理には高い計算コストを要する．提案手法では，パッチは各断面に対する正方形の領域となるため，パッチ探索処理に要する計算コストを削減することができる．

3.3 再構築処理

本節では，提案手法の処理の最終段階である再構築処理について説明する．再構築処理では，乱流転写処理によって高解像度化された断面の流れデータを入力とし，3 次元の流れを再構築する．図 4 にその模式図を示す．シミュ

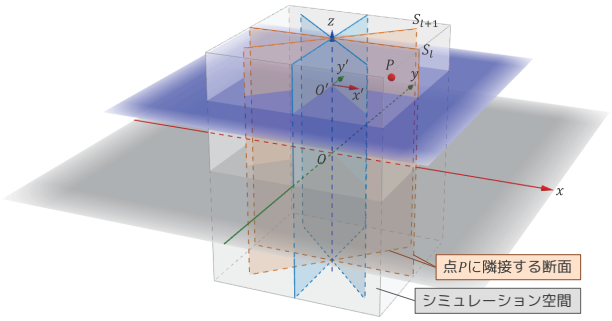

図 4: 再構築処理の模式図

レーション空間内のある点 P における流れの速度を再構築する場合を考える．シミュレーション空間の中心を原点とした場合の点 P の座標を (x, y, z) とおく．また，点 P を通り z 軸に垂直な平面における 2 次元座標系では，点 P の座標は (x, y) と表せる．この点 P を以下の式 (4) によって極座標 (r, θ) に変換する．

$$r = \sqrt{x^2 + y^2}, \theta = \arctan\frac{y}{x} \tag{4}$$

この偏角 θ を基に隣接する 2 枚の断面 S_l, S_{l+1} を特定する．l の値は，以下の式 (5) によって求められる．

$$l = \mathrm{floor}(\frac{\theta N_{slice}}{\pi}) \tag{5}$$

また，点 P を偏角方向に回転させたときに断面 S_l，S_{l+1} と交わる点をそれぞれ P_l, P_{l+1} とし，これらの点における断面の流れの速度を $\boldsymbol{v}_l$，$\boldsymbol{v}_{l+1}$ とする．点 P における速度 $\boldsymbol{v}_{2D}$ はこれらの速度から角度の線形補間により求められる．線形補間の模式図を図 5 に示す．点 P と点 P_l の偏角

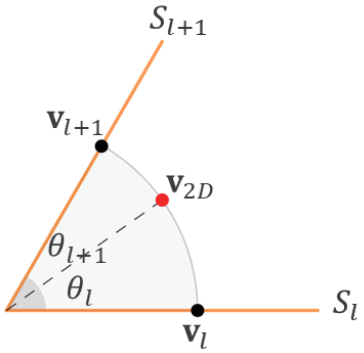

図 5: 再構築処理における線形補間

の差を θ_l とし，点 P_{l+1} と点 P の偏角の差を θ_{l+1} とすると，$\boldsymbol{v}_{2D} = (u, v)$ は以下の補間式 (6) によって求められる．

$$\boldsymbol{v}_{2D} = \boldsymbol{v}_l \times \frac{\theta_{l+1}}{\theta_l + \theta_{l+1}} + \boldsymbol{v}_{l+1} \times \frac{\theta_l}{\theta_l + \theta_{l+1}} \tag{6}$$

最後に，以下の式 (7) によってシミュレーション空間における 3 次元の速度 $\boldsymbol{v}_{3D}$ に変換する．

$$\boldsymbol{v}_{3D} = u\boldsymbol{n}_u + v\boldsymbol{n}_v \tag{7}$$

ここで，$\boldsymbol{n}_u$ は点 P の動径方向の単位ベクトルであり，$\boldsymbol{n}_u = (\cos\theta, \sin\theta, 0)$ と表され，$\boldsymbol{n}_v$ は z 軸方向の単位ベクトルであり，$\boldsymbol{n}_v = (0, 0, 1)$ と表される．

4. 実験結果と考察

本節では，提案手法を用いて作成した煙の例を結果として示す．実験に用いた計算機環境は，CPU が Intel(R) Core(TM) i9-9900KF(3.60GHz)，メモリが 16.0GB，GPU が NVIDIA TITAN RTX となっている．また，ターゲットとして格子数 $24 \times 24 \times 32$，格子幅 0.125 の流れを，ソースとして格子数 $192 \times 192 \times 128$，格子幅 0.015625 の流れを用い，結果として格子数が $192 \times 192 \times 256$，格子幅 0.015625 の流れを出力する．つまり，本実験ではターゲットを 8 倍に高解像度化する．いずれの流れもフレーム数は 200 とする．佐藤らの従来手法では，パッチサイズを $16 \times 16 \times 16$ とし，パッチ数は $12 \times 12 \times 16 = 2304$ となる．提案手法では，パッチサイズを 16×16 とし，パッチ数は断面当たり $12 \times 16 = 192$ となる．出力した流れに従って移流した密度場を用いて，3DCG 制作ソフトウェアである Maya によりレンダリングを行った画像を図 6 に示す．図 6 は，提案手法の分割数を変えて実験した結果と佐藤らの従来手法による結果を示す．また，これらの流れの生成にかかった計算時間を表 1 に示す．図 6 から，分割数が 4，8 の時には

表 1: 1 フレームあたりの計算時間の比較

分割数	4	8	16	32	64	従来手法
計算時間 [秒]	0.28	0.39	0.61	1.10	2.23	28.56
削減率 [%]	99.0	98.6	97.9	96.1	92.2	0.0

煙に縞模様が見えることが分かる．これは，分割数が小さいと煙の外側の流れが再構築処理における線形補間の影響を大きく受けるためであると考えられる．また，分割数が 16〜64 の時にはそのような縞模様が消えており，より詳細な乱流成分が付加されていることが分かる．これは，分割数を上げることで断面同士の距離が近くなり，再構築処理における線形補間の影響が小さくなるためであると考えられる．従来手法との見た目を比較すると，従来手法のほうがより詳細な乱流成分が転写されていることが分かる．これは，提案手法では流れを断面に分割し，再構築するという処理の中で流れの断面の法線方向の速度成分を失っているためと考えられる．さらに，表 1 から，分割数が増えると計算時間も大きくなっていることが分かる．しかし，分割数を 64 まで増やした場合でも，佐藤らの従来手法と比較して約 12.8 倍高速に煙を作成することができている．

5. まとめと今後の課題

本研究では，2 次元の乱流転写の活用により，佐藤らの乱流転写手法と比較してより効率的に高解像度の煙を作成する手法を提案した．従来手法では 3 次元空間において乱流転写を行っていたが，提案手法では入力の流れデータから放射状に断面を取得し，2 次元平面において乱流転写を行った．実験結果から乱流が転写された煙を約 12.8 倍高速に実行できることを確認した．今後の課題として，手法の汎用性を向上させることが挙げられる．今回は風の影響を無視して鉛直方向に立ち昇る煙のみを考慮した手法を提案したが，実際には風に煽られて煙の中心軸が歪むような場合があるため，そのような回転対称でない煙にも対応できるような手法が求められる．また，提案手法で作成した煙の視覚的精度を数値評価する手法が無いため，人間の知覚を反映した定量的な評価手法も検討する必要がある．

参考文献

[1] S. Sato, Y. Dobashi, T. Kim, and T. Nishita: *Example-based Turbulence Style Transfer*, ACM Trans. Graph., Vol. 37 (4), Article No. 84 (2018).
[2] N. Rasmussen, D. Q. Nguyen, W. Geiger, R. Fedkiw: *Smoke Simulation For Large Scale Phenomena*, ACM SIGGRAPH '03, pp.703-707 (2003).

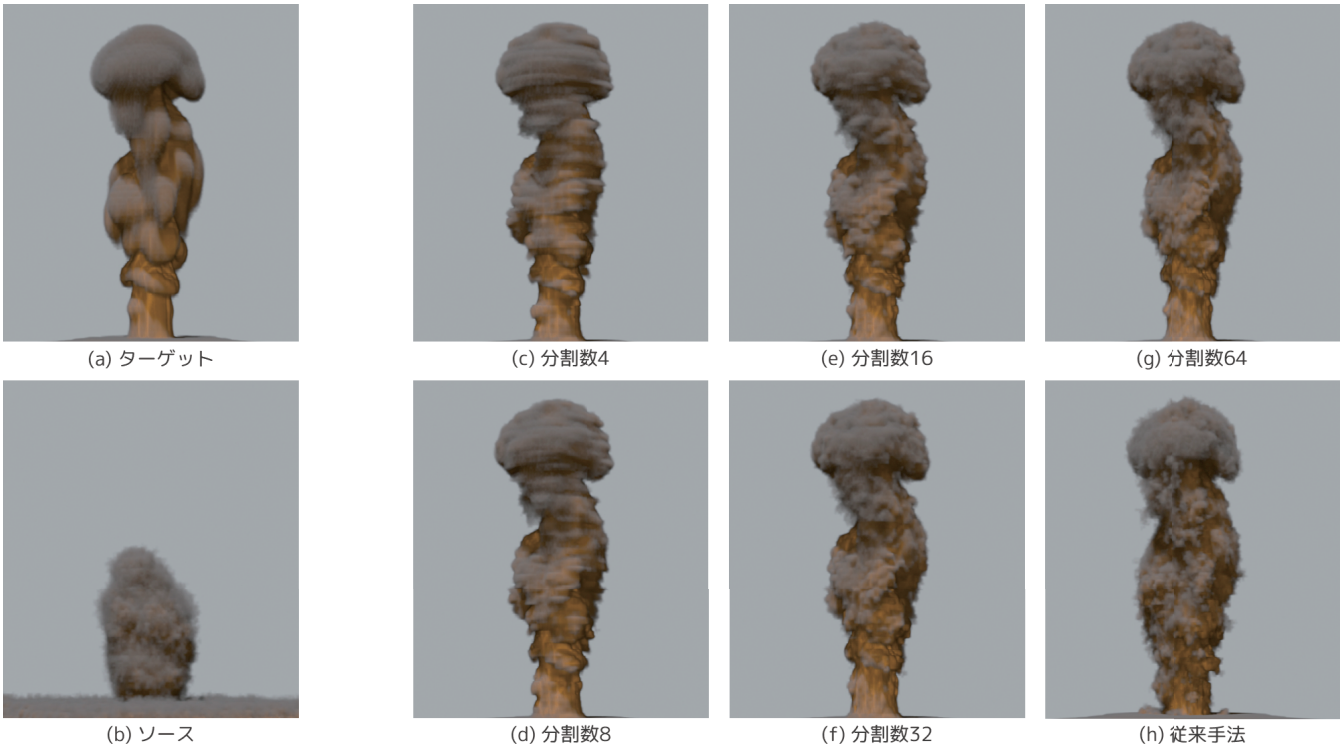

図 6: 分割数による煙の見た目の変化

[3] T. Kim, N. Thurey, D. James, M. Gross: *Wavelet turbulence for fluid simulation*, ACM Transactions on Graphics (SIGGRAPH North America) (2008).

[4] T. Pfaff, N. Thurey, A. Selle, M, Gross: *Synthetic Turbulence using Artificial Boundary Layers*, ACM SIGGRAPH Asia (2009).

[5] M. B. Nielsen, B. B. Christensen, N. B. Zafar, D. Roble: *Guiding of smoke animations through variational coupling of simulations at different resolution*, SCA ’ 09: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.217-226 (2009).