

アダプティブなスタイル転写による乱流合成

笹 晃輔^{1,a)} 佐藤 周平^{2,b)} 土橋 宜典^{1,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、視点からの流体の可視性に基づいて、従来法における計算領域を適応的に削減することで高速化を図る。煙の密度からシミュレーション空間中の各局所領域について視点への寄与を計算し、それに応じて各領域における従来法の合成処理の要否を決定する。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4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

Adaptive Example-based Turbulence 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. In this paper, we improve the previous method by adaptively choosing the computation domain based on the visibility of fluids from a viewpoint. We calculate a contribution to the viewpoint for each local regions in a simulation space using a density of smoke, and then determine whether the previous method need to be applied or not to each region. Our method achieves approximately four 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おいて、物理ベースのシミュレーションにより作成された流体の映像が映画やビデオゲーム等、広く利用されている。しかし、物理ベ

スのシミュレーションは計算コストが非常に高いことが問題の一つとなっている。また、所望の映像効果を表現するためには、シミュレーションに関するパラメータを適切に設定する必要があるが、そのためには何度もシミュレーションを繰り返し、適切なパラメータを見つける必要がある。これには高コストなシミュレーションの繰り返しが必要であり、膨大な手間と時間を要する。

上記の問題を解決するために、大域的な流れ（低周波成分）と詳細な流れ（高周波成分）を独立に計算する手法がいくつか提案されている [1], [2], [3], [4]。これらの手法で

¹ 北海道大学
Hokkaido University

² 富山大学
Toyama University

a) sasa@ime.ist.hokudai.ac.jp

b) ssato@eng.u-toyama.ac.jp

c) doba@ime.ist.hokudai.ac.jp

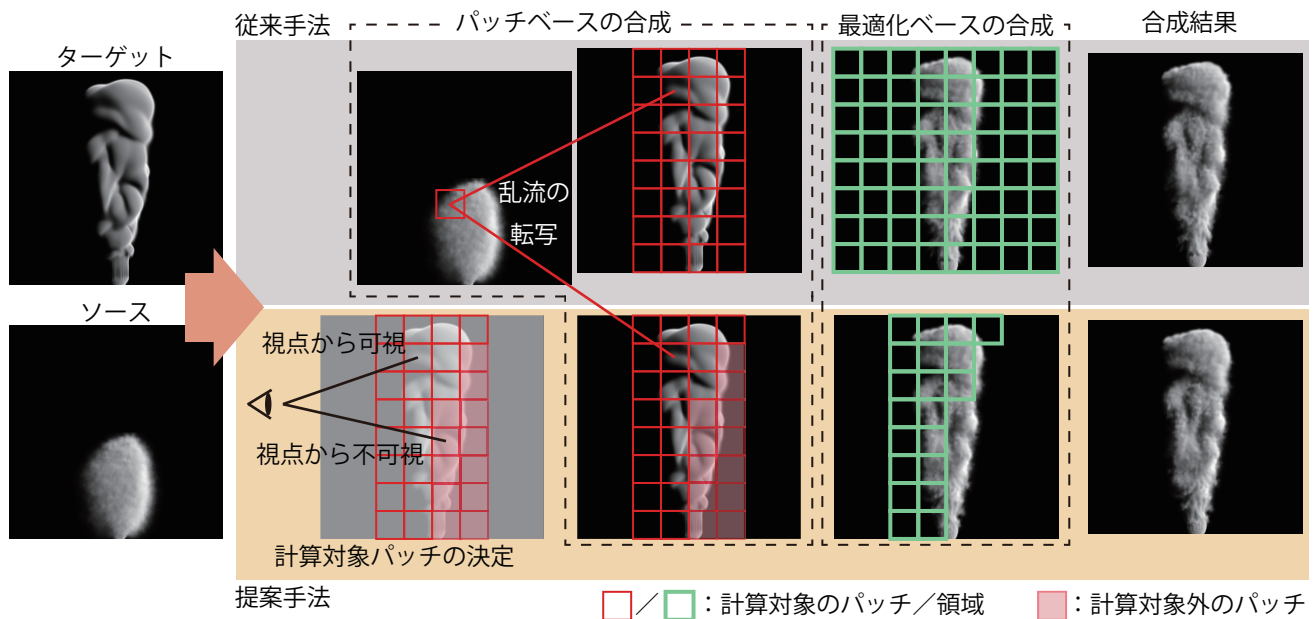


図 1: 従来手法および提案手法の概要

は、大域的な流れを低解像度のシミュレーションで設計し、その結果に対して詳細な流れを付加することで高解像度の結果を作成する。そのため、大域的な流れを保持しながら詳細な動きを付加することができ、所望の流れの作成にかかる作業負担を大幅に軽減できる。しかし、これらの手法では、依然として詳細な流れに関するパラメータの調整が必要である。流体の視覚的な印象は詳細な乱流成分に依存するところが大きく、また、パラメータから結果を予測することが難しいため、何度も試行錯誤が必要となる。

これに対し Sato らは、既に計算されている流れデータの高周波成分を再利用し、例示ベースで直観的に詳細な乱流を合成する手法を提案した [5]。Sato らの手法では、詳細な動きを持つ流れ（ソース）と低解像度のシミュレーションによりユーザがデザインした流れ（ターゲット）を入力として用意する。シミュレーション空間をパッチと呼ばれる小領域に分割し、パッチ単位でソースの低周波成分とターゲットとの類似した個所を探索する。そして、探索結果に基づきソースの高周波成分をターゲットへ転写する。この際、最適化ベースのテクスチャ合成手法を応用することで、パッチ間の不連続性を緩和している。Sato らの手法によってパラメータの調整が不要となり、より直観的な乱流合成が可能となるが、計算コストの高い処理を含んでおり、事前計算データのみを用いるにも関わらず単純にシミュレーションを実行したときと比較しても計算コストの削減率は小さい。特に計算コストが大きい処理は、パッチベースの探索・合成処理と、最適化ベースのテクスチャ合成によるパッチ間境界の平滑化処理である。

そこで本研究では、視点からの流体の可視性に基づく適応的なアプローチにより Sato らの手法 [5] を高速化する方法を提案する。すなわち、視点からの煙の見た目に寄与す

る部分のみに Sato らの手法を適用する。本稿では Sato らの手法と同様に流体として煙を扱い、視点から可視／不可視な煙の領域を判定する。具体的には、パッチにより分割されているシミュレーション空間に対し、各パッチ内に存在する煙の密度から透過率を計算し、そのパッチの煙が視点からの煙の見た目に影響するかどうかを判定する。そして、影響すると判定されたパッチについてのみ Sato らの手法を適用する。これにより、全体に適用した場合と同等の結果を得ながら、計算領域を削減でき、従来よりも短い時間で乱流転写が可能である。

2. 関連研究

流体の映像生成において、乱流の表現は映像のリアリティを向上させるために重要な要素である。しかし、乱流を表現するためには、高解像度・高精度の計算が必要とされる。そのため、後処理として乱流を合成する研究が数多く行われてきた。Kim らは低解像度の格子によりシミュレーションを行った結果に対し、ウェーブレットノイズにより生成した大小さまざまな渦を用いて欠落している乱流成分を合成する手法を提案した [1]。Pfaff らは流体と物体との相互作用により発生する乱流に着目し、流れによる境界層の分離をシミュレーションすることで乱流が発生する位置を特定し、再現する手法を提案した [2]。Nielsen らは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ための手法を提案した [3], [4]。しかし、いずれの研究も合成後の乱流の視覚的な印象はすべての処理が完了するまで確認できないという問題が残されている。そこで、Sato らはスタイル転写を応用した乱流合成手法を提案した [5]。これは事前に計算された高解像度の流れから詳細な乱流成分のみを抽出して別の流れに転写する手法

であり、あらかじめ転写に必要な乱流を含む流れも入力として用いるため、合成後の乱流の視覚的な印象を予測することができる。また、所望の乱流を含む流れを一度設計すると、その乱流成分を複数の異なる流れに繰り返し転写できるというメリットがある。

本研究と類似の考え方にに基づき、上西らは Kim らが提案した乱流合成手法 [1] を高速化する方法を提案した [6]。上西らは、視点から不可視となる領域や、視界に影響がない領域を各格子点毎に判定し、その領域の計算を削減することで、計算の高速化に成功している。しかし、Kim らの手法は元々すべての処理の計算コストが高くないため、計算時間の削減率は約 27 %にとどまっている。本研究では、上西らと類似の考え方にに基づき、視点への影響を考慮した計算領域の削減を行うことで、Sato らの手法を高速化する。

3. 提案手法

本研究では Sato らによる乱流転写手法 [5] を高速化することが目的である。そのため、まず Sato らの手法について説明し、その後我々の提案手法について述べる。Sato らの手法の概要を図 1 の上段に示す。この手法は入力として低解像度の流れ（ターゲット）と高解像度の流れ（ソース）を用意する。そして、これらに対して 2 段階の合成処理を経て高解像度の流れを得る。第 1 段階のパッチベースの合成では、ターゲットをパッチと呼ばれる小領域（図 1 赤色矩形）に分割し、内部に煙が存在しているパッチについてソースと類似しているパッチを探索し、ソースの高周波成分をターゲットへ転写する。第 2 段階の最適化ベースの合成では、パッチベースの合成において発生するパッチ境界の不連続を解消するため、最適化ベースのテクスチャ合成手法を応用した最小化問題をパッチ境界の領域（図 1 緑色領域）において解く。これらの処理を毎フレーム実行し、ソースの乱流が転写された高解像度の流れを得る。

次に、提案手法の概要を図 1 下段に示す。Sato らの手法では合成処理を煙が存在するパッチ全体に適用するのに対し、視点からの各パッチの可視性に基づいて適用するパッチを限定する。すなわち視点からの煙の見た目に影響のないパッチ（スキップパッチ）を判定し、そのパッチに対しては合成処理を適用しない。スキップパッチの決定は次のように行う。各フレームの最初に低解像度の流れ（ターゲット）において、各パッチに対し煙の密度から透過率なる値を計算し、その透過率がある閾値未満のパッチをスキップパッチとする。スキップパッチ部分の高周波の流れについては合成がなされないため、前フレームの結果を移流した流れを利用する。以下では、スキップパッチの計算方法と合成処理の削減についてそれぞれ詳細を述べる。

3.1 各パッチの可視性判定のための透過率計算

視点から煙のレンダリングを行う際、煙の内部に入射し

た光は散乱により減衰していき視点にまで届かなくなる領域が存在する。煙の濃さや厚さが大きくなるほど光が届かなくなる領域は増加していく。そのため、我々は光が減衰によって届かなくなる領域においては乱流の合成を行う必要がないと仮定し、以下の光の減衰計算に基づいて各パッチの透過率を求めることで、その領域を判定する。

煙などの関与媒質に入射した光は媒質による散乱や吸収によって減衰していく。視点位置を $\mathbf{x}_v$ 、格子毎の散乱光を $L_s(\mathbf{x})$ 、透過率を T_r とすると、視点 $\mathbf{x}_v$ から視線方向 ω に存在する距離 t の格子の位置と、視点に入射する光 $L(\mathbf{x}_v, \omega)$ はそれぞれ以下の式で表される。

$$\mathbf{x}(t) = \mathbf{x}_v + \omega t, \quad (1)$$

$$L(\mathbf{x}_v, \omega) = \int_0^T T_r(\mathbf{x}(t)) L_s(\mathbf{x}(t)) dt, \quad (2)$$

ここで、 T は視点からシミュレーション領域の奥行までの距離となる。透過率は視点から関与媒質内を進むに従って値が小さくなっていく。格子点に存在する煙密度を ρ 、減衰係数を σ とすると、視点から各格子点までの透過率を求める式は次のようになる。

$$T_r(\mathbf{x}(t)) = \exp(-\sigma \int_0^t \rho(\mathbf{x}_v + \omega s) ds). \quad (3)$$

透過率は視点 $\mathbf{x}_v$ から格子点 $\mathbf{x}(t)$ までの格子の密度 ρ を積分したものに、減衰係数 σ を乗算したものを指数として計算する。図 2 に視点位置 $\mathbf{x}_v$ 、視線方向 ω および格子点 $\mathbf{x}$ の関係を示す。減衰係数 σ は、最終的な結果のレンダリング時に設定される減衰係数と同じ値に設定する。また、式 (3) 中の t は視点位置から対象の格子点までの距離を表す。

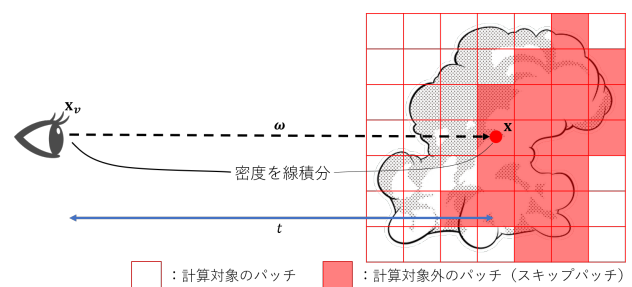
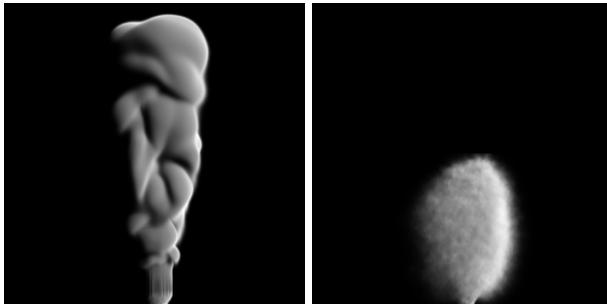


図 2: 透過率計算の模式図

本手法では、各格子点における透過率を利用してパッチの透過率を定義する。ただし、パッチは複数の格子点からなるため、すべての格子点に対して透過率計算をすると計算コストが高い。そのため、パッチ内のいくつかの代表点でのみ透過率を計算し、その中の最小値をそのパッチの透過率として設定する。代表点は、3 次元の場合、パッチの各頂点と各面の中心、そしてパッチの中心の計 27 点とした。



(a) ターゲット

(b) ソース

図 3: 入力に用いる流れ

3.2 パッチの可視性に基づく合成処理の削減

計算した各パッチの透過率とユーザ指定の閾値とを比較し、閾値未満であればそのパッチをスキップパッチとする。スキップパッチについては、パッチベースと最適化ベースの合成処理を行わずスキップする。パッチベース合成に関しては、単純にスキップパッチについてパッチの探索処理を行わず、転写する高周波成分が0の状態となる。最適化ベースの合成は、パッチ探索と連立一次方程式を解くための反復計算という2つの高コストな処理を繰り返し実行するため、Sato らによる手法の計算時間の大部分を占めている。これに対し、パッチ探索については上述のパッチベース合成と同様に探索処理を単純に実行しない。一方、従来法の連立一次方程式の計算では、煙の有無に関係なくパッチの境界となる格子全体について共役勾配法を用いて解を求めている。我々はこの連立一次方程式を解く領域についても煙が存在し、かつスキップされないパッチの境界のみに限定する。ただしこの場合、計算領域が毎フレーム変化するため、共役勾配法により解く際の係数行列を都度再構築する必要があり、計算コストが高くなる可能性がある。しかし、我々の実験から、再構築に要する時間よりも計算領域の削減の効果が大きく、全体として大きな高速化を達成することができた。結果の詳細については次節にて示す。

4. 実験結果と考察

提案手法の有効性を確認するため、従来手法との結果の比較を行う。主に、生成される結果に視覚的な差異がないこと、および計算コストが低減されていることを本比較実験により確認する。実験を行う計算環境は、Intel(R) Core(TM) i7-4770K 3.50GHz の CPU、16GB のメモリ、また GPU として GeForce GTX TITAN を搭載した PC である。実験に用いる流れデータは Sato らの論文と同一のものを使用しており、低解像度の流れ（ターゲット）は格子数 $24 \times 24 \times 32$ である。この流れに、高解像度のデータから乱流を転写し、各方向の格子数を8倍とした $192 \times 192 \times 256$ の結果を作成する。ソースの流れには、格子数 $192 \times 192 \times 128$ の高解像度シミュレーションにより作成したものをを用いる。

ターゲットに対して各パッチの透過率計算を行い、閾値処理によりスキップパッチを決定した結果を図4に示す。ここではシミュレーション空間を中央で切断した場合の2次元平面を可視化し、各画像の左中央に視点を配置している。それぞれの結果を見ると、スキップパッチの存在は煙密度の分布に依存しており、閾値が大きくなるにつれてその数が増加している。透過率は0.0~1.0の値を取り、煙内を進むにしたがって値は小さくなる。したがってこの例では、視点が存在する左側から右側に進むにつれ透過率の値が小さくなるので、右側のパッチほどスキップパッチとして判定されやすくなる。

次に、従来手法とスキップパッチによる計算領域の削減を行った提案手法における出力結果の比較を図5に示す。結果を比較すると、閾値が高くなるにつれて従来法との見た目の差異が目立つ結果となった。この原因としては、図4に示されるように、閾値が0.95の場合などは煙の表面を除くほぼ全てのパッチがスキップパッチとして決定されていることが考えられる。しかし、閾値が低すぎると削減領域がなくなるため、適切な閾値を設定することが今後の課題としてあげられる。

最後に、従来手法と提案手法における計算時間の比較を表1に示す。閾値0.85, 0.9, 0.95いずれの場合も削減率は75%以上となっており、特に最適化ベース合成における計算時間の削減が顕著になされていることが分かる。

5. まとめと今後の課題

我々は、高解像度化領域の削減を行うことで Sato らの手法の高速化する手法を提案した。従来手法では、パッチベースの合成が煙の存在する全パッチに対して適用されており、また最適化ベースの合成が全パッチ境界に対して適用されていることから計算コストが高かった。これに対し透過率の計算と閾値処理による領域削減を行った。実験結果から、従来手法と大きな見た目の差がない結果を約4倍高速に生成できることを確認した。

今後の課題として、様々な流れデータにおいて実験を行い、本手法の有効性および安定性を確認することがあげられる。また、障害物など煙以外の物体により遮蔽される領域がある場合に、その領域を判定し処理を削減することで、シーンによってはさらなる高速化が実現できる可能性がある。さらに、本実験では閾値をユーザ指定によるものとしていたが、見た目のクオリティと計算コストの両面から適切な閾値を自動で決定する手法の開発も挙げられる。

参考文献

- [1] Theodore Kim, Nils Thuerey, Doug James, and Markus Gross: *Wavelet turbulence for fluid simulation*, ACM Trans. Graph., Vol. 27 (3), Article No. 3 (2008).
- [2] T. Pfaff, N. Thuerey, A. Selle, and M. Gross: *Synthetic turbulence using artificial boundary layers*, ACM Trans.

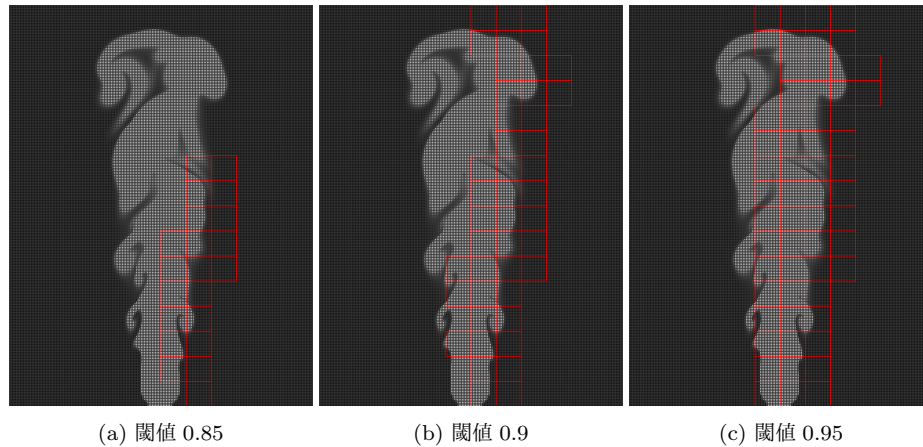


図 4: 算出されたスキップパッチ（断面を 2 次元で可視化）

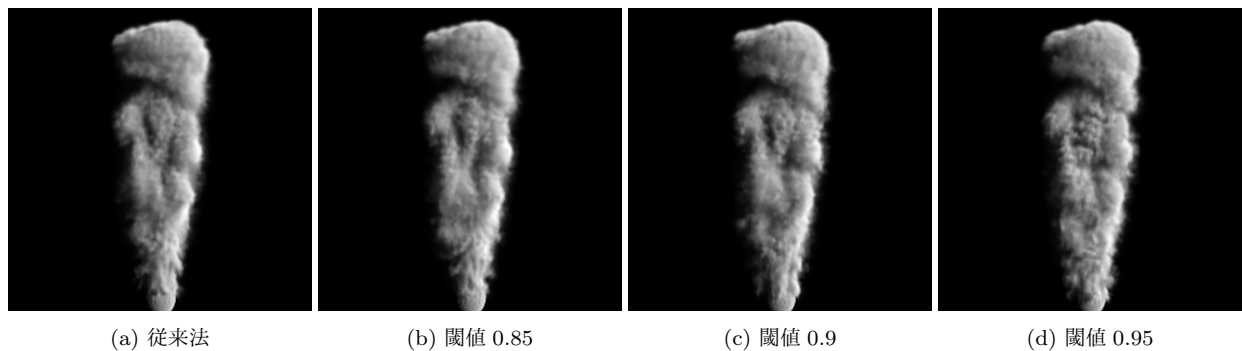


図 5: 従来手法および閾値を変えた場合の提案手法の結果の比較

表 1: 従来手法と提案手法における 1 フレーム当たりの平均計算時間の比較. T_p はパッチベース合成に要する時間, T_s は最適化ベースの合成に要する時間である.

	従来手法	閾値 0.85	閾値 0.90	閾値 0.95
T_p (秒)	0.83	0.67	0.65	0.61
T_s (秒)	27.73	6.39	5.37	4.52
合計 (秒)	28.56	7.06	6.01	5.13
削減率 (%)		75.26	78.95	82.03

学会第 79 回全国大会, pp.63-64 (2017).

- Graph., Vol. 28 (5), Article No. 121 (2009).
- [3] Michael B. Nielsen, Brian B. Christensen, Nafees Bin Zafar, Doug Roble, and Ken Museth: *Guiding of smoke animations through variational coupling of simulations at different resolutions*, Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 217–226 (2009).
 - [4] Michael B. Nielsen, and Brian B. Christensen: *Improved Variational Guiding of Smoke Animations*, Computer Graphics Forum, Vol. 29 (2), pp. 705–712 (2010).
 - [5] Syuhei Sato, Yoshinori Dobashi, Theodore Kim, and Tomoyuki Nishita: *Example-based Turbulence Style Transfer*, ACM Trans. Graph., Vol. 37 (4), Article No. 84 (2018).
 - [6] 上西 直人, 佐藤 周平, 岩崎 慶, and 西田 友是: 領域削減の自動化による *Wavelet Turbulence* の効率化, 情報処理