

## スタイル転写による乱流合成の高速化に関する一検討 A Study on Fast Example-based Turbulence Style Transfer

笹 晃輔<sup>†</sup>      佐藤 周平<sup>‡</sup>      土橋 宜典<sup>†</sup>  
Kosuke Sasa    Syuhei Sato    Yoshinori Dobashi

### 1. はじめに

近年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。

この問題を解決するため、既存の流れデータの高周波成分を再利用し、例ベースで直観的に詳細な乱流を合成する手法が提案されている[1]。こ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。

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 らの手法を高速化する方法を提案する。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。これにより、従来法に比べて約 4 倍の高速化を達成した。

### 2. 提案手法の概要

本研究では Sato らによる乱流転写手法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において解く。

次に、提案手法の概要を説明する。Sato らの手法では合成処理は煙が存在するすべてのパッチに適用するのに対し、我々の手法では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選択する。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

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。

#### 2.1 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が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  $\mathbf{x}_v$ 、格子点毎の散乱光を  $L_s(\mathbf{x})$ 、透過率を  $T_r$  とすると、視点  $\mathbf{x}_v$  から視線方向  $\omega$  に存在する距離  $t$  の格子点の位置と、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きまでの距離となる。透過率は視点から関与媒質内進むに従い値が小さくなっていく。格子点に存在する煙密度を  $\rho$ 、減衰係数を  $\sigma$  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。

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

透過率は視点  $\mathbf{x}_v$  から格子点  $\mathbf{x}(t)$  までの格子点の密度  $\rho$  を積分したものに減衰係数  $\sigma$  を乗算したものを指数として計算する。減衰係数  $\sigma$  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。

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。

#### 2.2 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

<sup>†</sup> 北海道大学 Hokkaido University

<sup>‡</sup> 富山大学 Toyama University

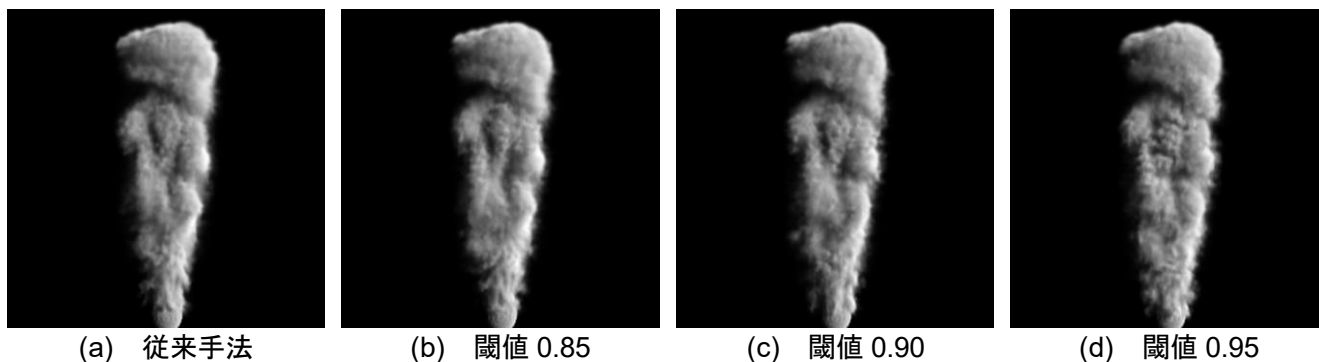


図 1 従来手法および閾値を変えた場合の提案手法の結果の比較

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。

### 3. 結果

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。

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 1 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、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。

### 4. おわりに

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。透過率の計算と閾値処理による領域削減を行うことで、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挙げられる。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。

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

### 参考文献

- [1] Syuhei Sato, Yoshinori Dobashi, Theodore Kim, and Tomoyuki Nishita, "Example-based Turbulence Style Transfer", ACM Trans. Graph., Vol. 37 (4), Article No. 84 (2018).