

例示ベース乱流合成のためのパッチベース探索の高速化

渡辺優太^{†1} 佐藤周平^{†2} 土橋宜典^{†1,2} 山本強^{†1}

概要 : CG 映像制作において、煙や水、炎などの流体の写実的かつ所望の映像表現を効率的に得るため、流体の低周波成分と高周波成分を独立して計算する手法が提案されている。我々も低解像度のシミュレーションにより生成した低周波の流れ場データ（ターゲット）に高解像度のシミュレーションにより生成された流れ場データ（ソース）の高周波成分を合成する手法を研究している。この手法ではターゲットとソースとの類似した領域をパッチベースで探索するが、全探索を行っているため計算コストが高い。そこで、本稿では、近似的な探索方法を導入し、その有効性について、精度と効率の面から検討を行ったので、その結果について報告する。

キーワード : 流体シミュレーション, 高解像度化, 高速探索

Acceleration of Patch-based Retrieval for Example-based Turbulence Synthesis

YUTA WATANABE^{†1} SYUHEI SATO^{†2} YOSHINORI DOBASHI^{†1,2}
TSUYOSHI YAMAMOTO^{†1}

1. はじめに

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 TV ゲーム等、多くのコンピュータグラフィックスアプリケーションに広く利用されている。しかし、物理ベースのシミュレーションは計算コストが非常に大きいことが問題の一つとなっている。また、所望の結果を得るためには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、その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、膨大な時間を要する。

そこで我々は、事前に計算された流れ場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ための方法の開発を行っている。映像制作現場では流体映像を作成する過程で数多くのデータが作成される。本研究ではそういったデータを有効活用し、流れ場を効率的かつ直観的に作成することを目的としている。我々のシステムでは、詳細な動きを持つ高解像度の流れ場（ソース）と低解像度の流れ場（ターゲット）を入力として用意する。そして、ターゲットとソースの低周波成分が類似した個所をソースの中から探索する。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合成する。探索はパッチベースで行い、近傍の流れ場の分布が最も類似している個所の高周波成分のパッチを合成に用いる。また、流れの大域的な特徴と局所的な特徴の両方を捉えるために、2 段階の解像度でパッチを探索する。本稿ではこのアプローチが

上記の方法により、良好な結果が得られることは確認しているが、類似領域を探索するための計算コストが大きな問題として残っている。そこで、本稿では、流れ場の性質を利用した近似的な探索により高速化を行う実験を行った。計算精度と効率の両面から検討を行ったので、その結果について報告する。

2. 例示ベース乱流合成

本節では、文献[1]の例示ベース乱流合成手法について記述する。ここでは簡単のため、2 次元の流れ場を用いそれぞれが同じフレーム数である場合を考える。また以下では小文字および大文字のアルファベットをそれぞれ低解像度および高解像度の格子上で定義される場に対して使用する。

2.1 入力データ

文献[1]の手法では入力データとして、ユーザが低解像度のシミュレーションによりデザインしたターゲットの流れ場（速度場 $\mathbf{u}_t(i)$ および密度に基づくマスク場 $d_t(i)$: 格子サイズ $n_x \times n_y$, 格子幅 Δx ）と、所望の詳細な動きを持つソースの流れ場（速度場 $\mathbf{U}_s(i)$ および密度に基づくマスク場 $D_s(i)$: 格子サイズ $N_x \times N_y$, 格子幅 ΔX ）を使用する。ここで $i=0, 1, \dots, I-1$ でありフレームカウントを表し、 I は入力の流れ場のフレーム数である。また、 $d_t(i)$, $D_s(i)$ はブール値が格納された場であり、シミュレーションを行った際の密度値が D_{th} よりも大きくなる格子点では 1, その他は 0 となる。 D_{th} は密度の閾値であり、煙の領域が抽出されるように決定する必要がある。

^{†1} 北海道大学
Hokkaido University
^{†2} UEI リサーチ (株式会社ドワンゴ)
UEI Research (DWANGO Co., Ltd.)

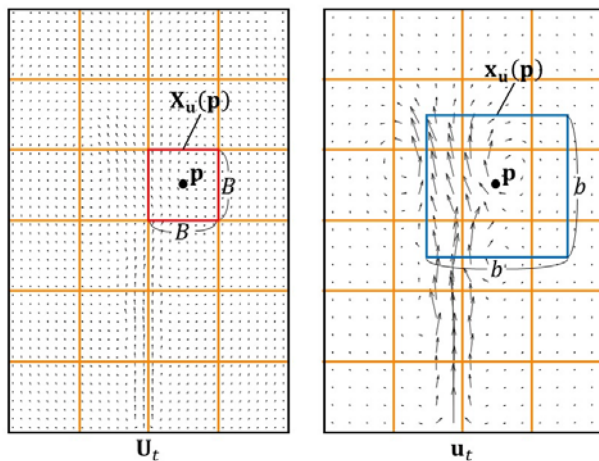


図 1 パッチへの分割
Figure 1 Division into patches.

次にターゲットの解像度 $n_x \times n_y$ と同一になるようソース $U_s(i)$, $D_s(i)$ をダウンサンプルすることで低周波成分 $u_s(i)$, $d_s(i)$ を得る。また, $u_t(i)$, $d_t(i)$ および $u_s(i)$ についてソースの解像度 $N_x \times N_y$ になるよう線形補間した場 $U_t(i)$, $D_t(i)$ および $U_{st}(i)$ を用意する。そして, ソースの高周波成分 $U_{sh}(i)$ を $U_s(i)$ と $U_{st}(i)$ との差分から算出する。文献[1]の手法ではフレームごとに独立に計算を行うため, 以降不要な部分では i を省略する。

次にターゲットの流れ場をパッチに分割する。 U_t (および D_t) について全体を $B \times B$ の小領域に分割する (図 1 左の橙色格子)。そして中心が p である小領域をそれぞれ $X_u(p)$, $X_d(p)$ と記述し, これらをローカルパッチとする。続いて各 p について u_t (および d_t) 上に p を中心とした $b \times b$ の領域を定義し (図 1 右の青色矩形), それらのパッチを $x_u(p)$, $x_d(p)$ とする。また, $x_u(p)$, $x_d(p)$ はグローバルパッチと呼ぶこととする。 b は $b > B \times \Delta x / \Delta x$ となるように設定する。

2.2 パッチベース探索の方法

本節では文献[1]の手法におけるパッチベースの探索方法について記述する。ターゲットに類似するソースの個所をパッチベースで探索し, ターゲットに合う高周波成分を合成する。この手法では流れ場の大域的な特徴と局所的な特徴を両方捉えるために 2 段階の解像度で探索を行う。

まず, 低解像度の格子を用いて大域的に類似個所を探索し (図 2 左), 次の局所的な探索における探索範囲を求める。2.1 節で得た各パッチについて, 以下のエネルギー関数 E_l を最小化するようなソース上の位置 q_l を求める (図 2 左)。

$$E_l(p, q_l) = \|x_u(p) - z_u(q_l)\|^2 + \alpha \|x_d(p) - z_d(q_l)\|^2 \quad (1)$$

ここで, α は 2 項目の影響を調整するための係数でありユーザにより指定される。また, $z_u(q_l)$, $z_d(q_l)$ は u_s , d_s について各格子点 q_l を中心とした $b \times b$ のサイズのパッチを表す。

次に, 高解像度の格子において上記で求めた位置 q_l を中心とした探索範囲において, 類似度の高い個所を局所的に探索する (図 2 右)。また, 時間的なコヒーレンスを考慮するため, 1 ステップ前に提案手法により合成された高周波成分 $U_h(i-1)$ を $U_t(i-1)$ に従って移流させた場 $U_h^*(i)$ を使用する。そして, 各パッチについて下記のエネルギー関数を最小化するソース上のパッチの位置 q_h を探索する (図 2 右)。

$$E_h(p, q_h) = \|X_u(p) - Z_u(q_h)\|^2 + \alpha \|X_d(p) - Z_d(q_h)\|^2 + \beta \|X_u^*(p) - Z_u^*(q_h)\|^2 \quad (2)$$

ここで, β は 3 項目の影響を調整するための係数でありユーザにより指定される。また, $Z_u(q_h)$, $Z_d(q_h)$ および $Z_u^*(q_h)$ は U_s , D_s および U_{sh} について各格子点 q_h を中心とした $B \times B$ のサイズのパッチを表す。

上記 2 段階の探索を行うことでターゲットの各パッチについて流れが最も類似するソースのパッチが得られ, それらを用いて結果の高周波成分 $U_h(i)$ を合成する。最後に $U_t(i)$ に $U_h(i)$ を加算して結果の流れ場 $U(i)$ を得る。

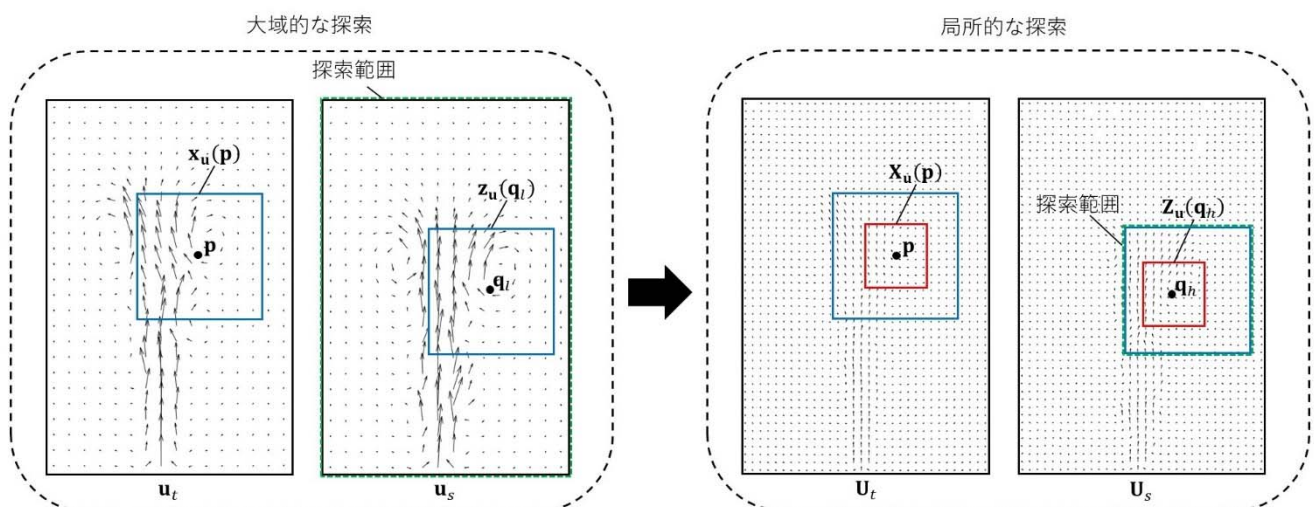


図 2 パッチベース探索の方法
Figure 2 The patch-based retrieval method.

3. アダプティブな探索

前節で述べた手法によって、良好な結果を得ることはできるが、計算時間に問題がある。現状では、類似パッチの探索は全探索によって行っているため、多大な計算コストがかかるためである。したがって、初めから高解像度のシミュレーションを行うよりも計算時間が長くなってしまいう場合が多い。一つの解決策として、例えば **kd-tree** などを用いて、ソースの全パッチを事前にクラスタリングすることで、類似パッチ探索を高速化することが考えられる。しかし、我々の実験では、この方法では事前計算および **kd-tree** を保存するためのコストが膨大となってしまう、非現実的であることが分かった。そのため、クラスタリングなどの事前計算を行うことなく、効率的に類似パッチを探索する方法が求められる。

本研究では、流れ場の空間的な類似性に着目し、事前計算なしに高速化する可能性を探る。隣接するパッチ内の流れ場は多くの場合、極めて類似しており、全探索を行うことは極めて効率が悪い。そこで、ソース内の考えられるすべてのパッチを探索するのではなく、探索対象領域内をアダプティブに探索する方法を検討する。具体的には、対象領域内を粗くサンプルして抽出したより少数のパッチの中から最も類似したパッチを探索する。そして、その周辺領域に対してより詳細に探索を行う。この操作を繰り返し行うことで、精度を保ちつつ探索の効率を向上できることが期待できる。

対象領域内に $(n \times n)$ のパッチが存在している場合を考える。まず、初期のサンプリング間隔 m_0 として、 $(n/m_0) \times (n/m_0)$ のパッチを等間隔に抽出し、最も類似したパッチを探索する。次に、見つかったパッチ周辺において、同様に等間隔にパッチをサンプルする。ただし、このときのサンプリング間隔は $m_i = m_0/2$ とする。この処理を繰り返し、 $m_i = 1$ (i は繰り返し回数) となった時点で処理を終了する。

4. 実験結果

前節で述べたアダプティブな探索方法の有効性について調査した結果を述べる。使用した計算機は Intel® Core™ i7-2600K (3.40GHz) を搭載したデスクトップ PC である。実験には 2 次元の流れ場データを使用し、局所的な探索について効率的な探索を試みた。ターゲットの流れ場に格子サイズ 16×24 、格子幅 0.25、ソースの流れ場に格子サイズ 128×192 、格子幅 0.03125 のデータをそれぞれ 200 フレーム分使用し、 $B = 16$ 、 $b = 4$ として実験を行った。

まず、予備実験として、単純に等間隔にサンプルした場合について、サンプリング間隔を変更した場合の誤差と効率について調査した。誤差については、全探索を行った場合との相対誤差の平均値と絶対誤差の平均値を使用した。結果を表 1 の 2~4 行目に示す。この結果から分かるよう

に、サンプル数を減らすことで探索時間を短縮することができるが、それに伴い探索の誤差も大幅に増加してしまう。

次に、アダプティブな探索について初期のサンプル間隔を 8 として実験を行った。結果を表 1 の最下段に示す。この結果から、アダプティブな探索により高い精度を保ったまま探索を約 9 倍高速化できていることが分かる。

また、アダプティブな探索について、フレームごとの平均相対誤差と平均絶対誤差をそれぞれ図 3、図 4 に示す。最終的な結果に影響するのはターゲットの流れ場の密度を含んだ部分であるため、ターゲットのパッチのうち密度の

表 1 探索方法ごとの平均誤差と 1 フレーム当たりの探索時間

Table 1 Average error and retrieval time per frame by the retrieval method.

探索方法	相対誤差	絶対誤差	探索時間(秒)
全探索		0	0.1145
サンプリング間隔: 2	0.2341	0.3824	0.0313
サンプリング間隔: 4	1.1060	1.2969	0.0098
サンプリング間隔: 8	3.0804	3.3836	0.0036
アダプティブな探索	0.0103	0.1935	0.0125

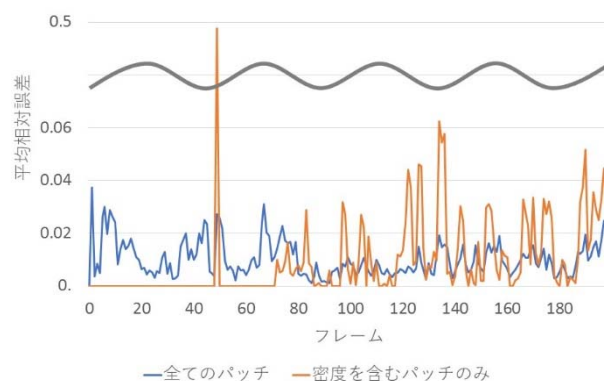


図 3 フレームごとの平均相対誤差

Figure 3 Average relative error by the frame.

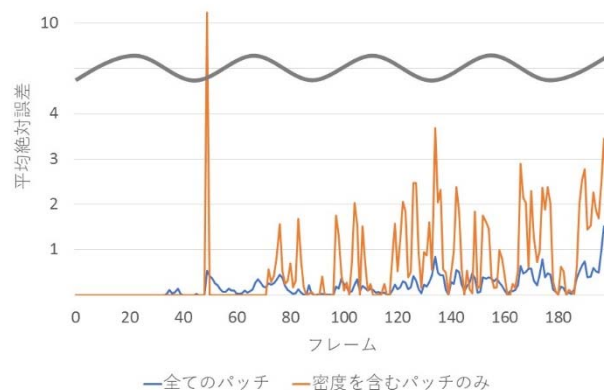
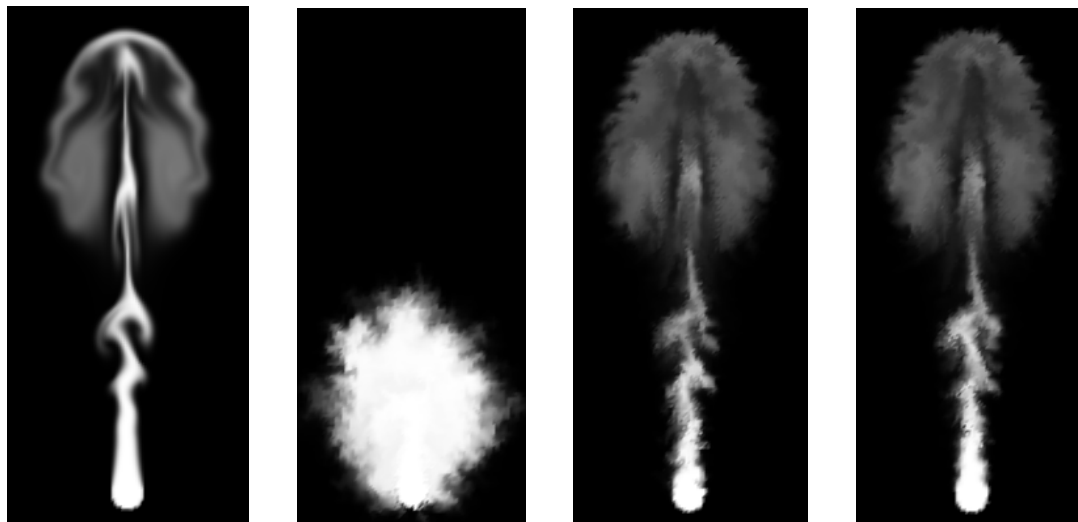


図 4 フレームごとの平均絶対誤差

Figure 4 Average absolute error by the frame.



(a)ターゲットの流れ場 (b)ソースの流れ場 (c)全探索による結果 (d)アダプティブな探索による結果

図 5 探索方法ごとの乱流合成の結果

Figure 5 The results of turbulence synthesis by the retrieval method.

マスクが 1 である格子を含んだものに絞って平均誤差を算出した結果も示す。図を見ると、49 フレーム目において密度を含むパッチの平均相対誤差及び平均絶対誤差が極端に大きくなってしまっている。このフレームでは、密度を含む 5 つのパッチのうち 1 つが、相対誤差 2.48、絶対誤差 51.2 と大きくなってしまっていた。同じフレームの他のパッチや前後のフレームについては高い精度で探索できていた。誤差が極端に大きくなってしまふことは望ましくないため、原因を調査する必要がある。

最後に、アダプティブな探索により探索したパッチを用いて 2 節で述べた方法により乱流合成を行った結果を図 5 に示す。図 5(a)はターゲットの流れ場、図 5(b)はソースの流れ場、図 5(c)は全探索による結果、図 5(d)は局所的な探索にアダプティブな探索を使用した結果を示している。アダプティブな探索でも全探索とほぼ変わらない見込みが得られており、探索の精度を保ちながら高速化できていることが分かる。

5. まとめと今後の課題

本稿では、例示ベース乱流合成手法で必要となる 2 次元の流れ場の類似パッチの探索について、アダプティブな探索方法により事前計算なしで約 9 倍の高速化を達成した。

今後の課題として、4 節で述べた誤差が極端に大きくなってしまふ場合について原因を調査する予定である。また、3 次元の流れ場に対する実験を行い、有用性を検証する予定である。

謝辞 本研究は JSPS 科研費 JP15H05924 の助成を受けたものです。

参考文献

- [1] 佐藤周平, 土橋宜典, 西田友是, 流体アニメーションのための例示ベース乱流合成. *Visual Computing / グラフィクスと CAD 合同シンポジウム* 2017, 2017.