

実写画像に類似した流体アニメーションの生成

古川 歩¹ 土橋 宜典¹ 山本 強¹

概要: CGによる自然現象の表現は重要な研究分野である。中でも、水や煙、炎といった流体現象のシミュレーションに関する研究に注目が集まっている。流体現象を表現する方法として、流体解析の方程式に基づいたシミュレーションが挙げられる。しかし、所望の結果を得るためには、多くのシミュレーションパラメータを試行錯誤的に調整しなければならず煩雑な作業が必要となるためユーザの負担が大きくなる。また、より写実的な結果を得るためには精度の高いシミュレーションを行わなければならない、計算コストが大きいという問題がある。そこで、本研究ではシミュレーションによって得られた温度分布に Texture Synthesis を適用することで実写画像と類似したアニメーションを効率的に生成する手法を提案する。

キーワード: 流体シミュレーション, Texture Synthesis

Creating Realistic Animation of Fluids using Photographs

AYUMU KOGAWA¹ YOSHINORI DOBASHI¹ TSUYOSHI YAMAMOTO¹

Abstract: Visualization of natural phenomena has become one of the most important research topics in computer graphics. In particular, visual simulation of fluid phenomena, such as water, smoke and fire, has attracted many researchers' attentions. One of the common approaches to the simulation of fluid phenomena is to numerically solve equations governing the fluid motion. However, in order to obtain desired visual effects, users have to adjust many simulation parameters by a time-consuming trial and error process. Furthermore, accurate and expensive numerical simulation is required to obtain realistic results. We propose a system that efficiently creates realistic animations by applying Texture Synthesis techniques to temperature distribution obtained by fluid simulation.

Keywords: Fluid Simulation, Texture Synthesis

1. はじめに

近年、コンピュータグラフィックス (CG) 技術の発達により、リアルな映像を生成することや現実では起こりえないことを表現することが可能となった。CG 技術は様々なシーンで使用され、その魅力を高める重要な要素の一つであり、中でもゲームや映画などのエンターテインメントの分野では幅広く用いられている。その中でも、自然現象を対象とした研究は盛んに行われ、風や煙、水、炎、雲などの表現に関する様々な手法が提案されてきた。本稿では、様々な自然現象の中で特に水や煙、炎といった流体現象

に着目し写実的にアニメーションを生成する方法を提案する。

流体映像を生成する方法はいくつかある。例えば、流体シミュレーションを用いる方法 [1][2][3] やビデオから流体の動きを抽出し、それを絵画などの静止画に合成する方法 [7] がある。流体シミュレーションを用いて映像を生成する場合、流体解析の方程式に基づいてシミュレーションを行う。しかし、シミュレーションの場合、数多くのパラメータに依存しているため所望の結果を得るためには流体解析に対する専門的な知識と複雑なパラメータの調整が必要である。また、よりリアリティのある流体映像を生成するためにはシミュレーションの解像度を上げて計算を行わなければならない計算コストが高くなるという問題がある。

¹ 北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology,
Hokkaido University

また、ビデオデータベースを用いて、静止画の流体をアニメーションする方法では、生成される動きがデータベース内のデータに限られるため、所望の動きを生成するためには、数多くのビデオデータが必要である。また、異なるビデオ映像から切り取った映像を組み合わせるため、不自然な映像が生成される場合がある。

本研究では、流体シミュレーションにより得られた画像に対し、実写の流体画像の詳細を付加することで、効率的に写実的な流体アニメーションを生成する手法を提案する。詳細の付加には、Texture Synthesis[8] と呼ばれる手法を適用する。提案手法では、まず低解像度シミュレーションにより所望の外形を持つアニメーションを作成する。そして、それに対して実写画像から流体の詳細な部分を Texture Synthesis を用いて付加する。しかし、単に Texture Synthesis を毎ステップ適用しただけでは、時間的に不連続となる。そこで、シミュレーション結果の速度場を用いることでこの問題の解決を図る。本稿では特に炎の流体現象について着目し、2次元の炎のシミュレーションを用いて実験を行った。

以降、本稿の構成は、2節で関連研究について簡単な紹介を行い、3節で提案手法について述べる。4節で提案手法を用いた実験結果とその考察について述べ、最後に5節でまとめとする。

2. 関連研究

流体のビジュアルシミュレーションに関する研究として Stam の手法 [1] が挙げられる。この手法は、流体解析の方程式である Navie-Stokes 方程式を解く際の安定性の問題を解決し、流体シミュレーションを実用化することに成功した。これをきっかけに、様々なビジュアルシミュレーションの研究が行われている。本稿は炎のシミュレーションに着目した。炎のシミュレーションに関する従来研究として、Nguyen らの手法 [2] や Kang らの手法 [3] が挙げられる。Nguyen らの手法では、燃料と燃焼物に対して個別に流体解析を行い、レベルセット法を用いてその相互作用を計算することで実際の物理現象に近い炎のシミュレーションを行う。また、Kang らの手法では格子法と粒子法によるシミュレーションを組み合わせ、炎などの化学反応を伴った流体现象を表現する。これらの手法によって写実的な炎のシミュレーションが可能となったが、計算コストが高いという問題がある。

効率的に流体映像を生成するための手法として、低解像度のシミュレーションを高解像度化するという手法が提案されている。Yuan らの手法 [4] では、低コストな低解像度のシミュレーションを用いて作成した流れ場を用いて、流体の動きを制御する手法を提案している。また、佐藤らの手法 [5] では、前計算により作成したデータベースを用いて低解像度の流体シミュレーションを高解像度化すること

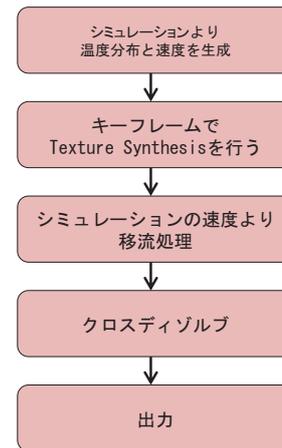


図 1 提案手法の概要

で、効率的に高解像度の流体アニメーションの作成を可能としている。前処理で得られた速度場をデータベースとして保存しておき、ユーザ所望の流れの場をデータベース内の速度場の線形和として表現している。また、ノイズを付加することで疑似的に流体の詳細を付加する手法 [6] も存在する。この手法では、乱流ノイズ関数と低解像度シミュレーションを組み合わせることで細部の情報を付加した高解像度映像を出力する。しかし、これらの手法では表現可能な炎のバリエーションには限りがある。提案法では、実写画像を基にアニメーションを生成するため様々な炎を表現できる。

シミュレーションを用いずに流体映像を生成する手法として Okabe らの手法 [7] がある。目的の静止画に対しビデオデータベースの一部を目的画像の一部に割り当てて合成することで任意の流体の静止画をアニメーションさせることができる。この手法は、異なるビデオ映像から切り取った局所的な映像を組み合わせるため不自然な映像が生成される場合がある。

本稿では、シミュレーション結果画像に Texture Synthesis[8] を用いることで効率よく目的画像と類似した流体アニメーションを生成する手法を提案する。

3. 提案手法

提案システムの概要を図 1 に示す。まず、3.1 節で述べる方法を用いて炎のシミュレーションを行い温度分布と速度場を取得する。次に、一定間隔に配置されたキーフレームに対して、シミュレーションの温度分布とユーザに指定される実写画像を用いて Texture Synthesis により詳細が付加された温度場を生成する。キーフレーム間については、シミュレーションで得られる速度場を用いたキーフレームでの温度場を移流させることにより生成する。最後に、よりスムーズな変化を表現するためにクロスディゾルブを行う。

以下、提案法の詳細を 3.1 節以降で説明する。

3.1 流体シミュレーション

文献 [1] は、安定かつ高速に流体の動きを計算する手法を提案している。従来の Navier-Stokes 方程式 (N-S 方程式) によるシミュレーション方法では計算を安定に行うためには、タイムステップを非常に小さくする必要があった。しかし、この手法では、N-S 方程式の移流項 (式 (1) 第 1 項) の計算にセミラグランジュ法を用いることで、タイムステップが大きい場合でも安定に計算できる近似解法を提案している。以下にこの手法で用いる N-S 方程式を示す。

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla p + \mathbf{f} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

ここで \mathbf{u} は、流体の速度ベクトルである。また、 ρ は流体の密度、 p は圧力である。 \mathbf{f} は外力を表しており、重力、風などの様々な影響を考慮できる。式 (1), (2) は、非圧縮性条件下における N-S 方程式であり、非圧縮性流体を扱う場合のみ式 (2) が成り立ち、この式は連続の式と呼ばれる。また、式 (1) は速度場に対する N-S 方程式であり、右辺第 1 項から、移流項、圧力項、外力項である。本研究では、炎を簡易的に考慮するため式 (1) により計算された速度場を用いて温度場の時間発展を計算する式 (3) を用いる。

$$\frac{\partial T}{\partial t} = -(\mathbf{u} \cdot \nabla) T - C_r \left(\frac{T - T_{amb}}{T_{max} - T_{amb}} \right)^4 \quad (3)$$

ここで、 T は温度、 C_r は冷却定数、 T_{amb} は環境温度である。また、 T_{max} はシミュレーション空間内での最大温度であり、シミュレーションにより得られた値を設定する。右辺第 2 項は流体温度の環境への放射損失を表している。また、浮力 \mathbf{f}_{buo} と渦補正力は \mathbf{f}_{conf} は以下の式で与えられる。渦補正力は、乱流成分を疑似的に付加するために用いられ Fedkiw らの手法 [9] に基づいている。

$$\mathbf{f}_{buo} = \kappa_b (T - T_{amb}) \mathbf{y} \quad (4)$$

$$\mathbf{f}_{conf} = \epsilon (\mathbf{N} \times \omega) \quad (5)$$

ここで式 (4) の κ_b は浮力係数、 \mathbf{y} は鉛直方向の単位ベクトルである。また、式 (5) の ϵ は渦補正係数で、 \mathbf{N} は $\omega = \nabla \times \mathbf{u}$ を用いて表現すると $\mathbf{N} = \nabla |\omega| / |\nabla |\omega||$ になる。この浮力と渦補正力は式 (1) の外力項に適用される。

本稿では格子法により流体解析を行う。2次元のシミュレーション空間を格子で分割し、各格子点において速度および温度を割りつける (図 2 参照)。そして、シミュレーション空間中央下端に熱源を配置し、温度と速度の 2 つの値の時間変化を上記で説明した方法を用いて計算する。

3.2 Texture Synthesis を用いた詳細の付加

実写画像から流体の詳細な部分を付加するために Texture Synthesis を用いる。具体的には、3.1 節で得られるシミュ

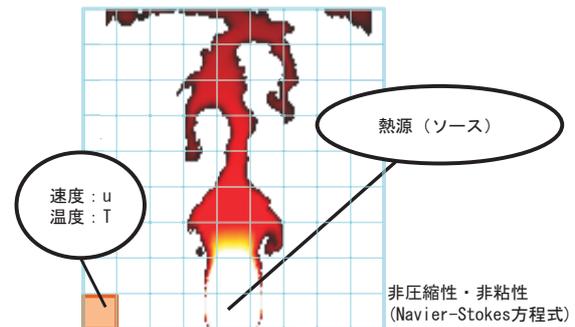


図 2 シミュレーション空間

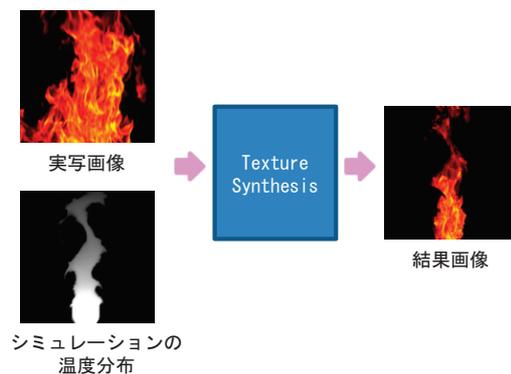


図 3 Texture Synthesis 適用例

レーションの温度分布と類似するように実写画像の詳細が付加された画像を新たに生成する。提案手法では、文献 [8] の User Control を用いた Texture Synthesis を適用した。この方法では、指定された目的分布に対してその各局所領域がユーザ指定の実写画像の局所領域と類似するように新しい画像を生成できる (図 3 参照)。具体的なアルゴリズムを説明するために、 \mathbf{I}_a をユーザ指定の実写画像、 \mathbf{I}_s を出力画像、 \mathbf{I}_t をシミュレーションの温度分布の濃淡画像、 p_i を入力画像の注目画素、 p を出力画像の注目画素、 $\mathbf{N}_U(p)$ を p の近傍画素領域 (図 4 に示す上の L 字領域)、 $\mathbf{N}_L(p)$ を p の近傍画素領域 (図 4 に示す下の L 字領域) と定義する。 $\mathbf{N}_U(p)$ は入力画像と出力画像の輝度値を比較し、出力画像の色を決定するための領域で、 $\mathbf{N}_L(p)$ はシミュレーションの温度分布と入力画像の α 値を比較し、出力画像の形状を決定するための近傍画素領域である。

初期設定としてホワイトランダムノイズで \mathbf{I}_s を生成し、Texture Synthesis を用いてこのノイズを修正していくことで、 \mathbf{I}_a と類似した \mathbf{I}_s を生成する。 \mathbf{I}_s の注目画素 p を決定するためにラスタ走査により出力画像の近傍画素領域 $\mathbf{N}_U(p)$ 、 $\mathbf{N}_L(p)$ と入力画像の近傍画素領域 $\mathbf{N}_U(p_i)$ 、 $\mathbf{N}_L(p_i)$ を比較する。最も類似した $\mathbf{N}_U(p_i)$ と $\mathbf{N}_L(p_i)$ を用いて入力画像の注目画素 p_i を出力画像の注目画素 p に割り当てる。入力画像と出力画像の近傍画素領域の類似性を計算す

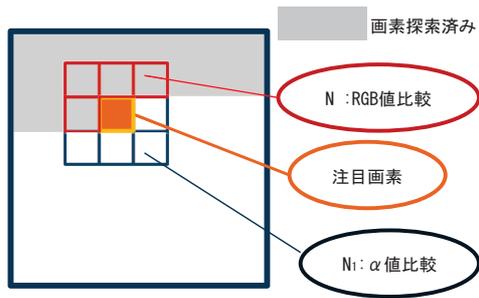


図 4 近傍画素群

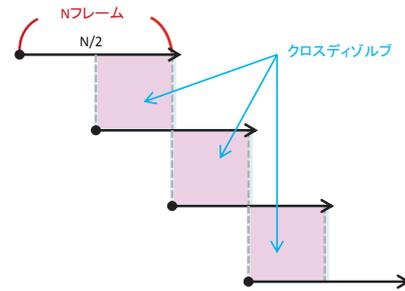


図 6 クロスディゾルブ

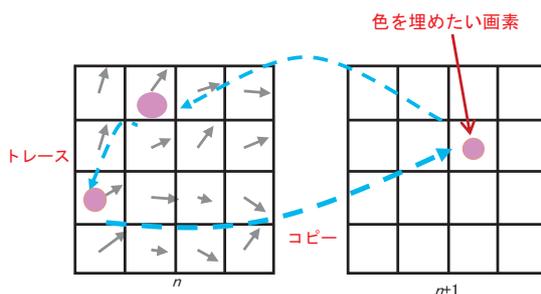


図 5 バックトレース

るために二乗誤差の総和を用いている。出力画像の全ての画素が埋まるまで繰り返し計算される。

3.3 シミュレーションの速度場の適用

各シミュレーション結果画像に対して Texture Synthesis を適用しただけでは不連続な映像が得られる。これは、時間的連続性を考慮していないためである。そこで、時間的連続性を考慮するためにシミュレーションの速度場を用いて連続した映像を生成する。シミュレーションの温度分布を用いて Texture Synthesis を適用し入力画像と類似した結果画像を作成した後に、シミュレーションの速度場を適用して新たに生成された結果画像の温度分布を移流させることにより、次のフレームを算出する。シミュレーションの速度場を用いて移流させる際には、バックトレースを用いた(図5参照)。

3.4 クロスディゾルブ

本手法では、よりスムーズな変化を表現するためにクロスディゾルブを用いる。まず、3.3節の処理を行い、Nフレーム分の結果画像を作成する。クロスディゾルブを用いて重ねるフレームをmフレームとし、重ねる先頭フレームの画像を Texture Synthesis を用いて作成する(図6参照)。

4. 実験結果

提案手法を適用した結果を示す。実験環境は、CPU: Intel Core i7 2600K, メモリ: 8.0GB, GPU: NVIDIA GeForce

GTX 560, グラフィックス API が OpenGL である。また、用いるシミュレーションの解像度は 128×128 である。

提案法を適用した結果を図7に示す。図7(a),(e),(i)は入力画像を表わし、図7(b),(f),(j)はシミュレーションの温度分布の濃淡画像であり Texture Synthesis を行うときのターゲットである。また、図7(c),(g),(k)は Texture Synthesis を適用した結果画像である。図7(d),(h),(l)はシミュレーションの速度場を用いてバックトレースを行い、画素を移流させた図7(c),(g),(k)の5フレーム後の結果画像である。結果画像を見てみると、シミュレーション結果画像がより入力画像と類似した写実的な結果が得られたことが確認できる。また、シミュレーションの速度を用いて移流させていることにより図7(c),(g),(k)は色がぼけたような結果が得られた。

5. まとめ

本稿では、シミュレーション結果画像に対して Texture Synthesis を適用することで入力画像と類似した写実的な映像を生成することを提案した。ユーザは入力画像とシミュレーションの温度分布を入力する。また、流体シミュレーションの速度場を用いることでリアルな炎のアニメーションを再現している。本稿では、2次元シミュレーションに対して実験を行ったが、今後3次元シミュレーションを用いて実験を行う予定である。

参考文献

- [1] J. Stam, "Stable fluids," Proc. ACM SIGGRAPH 1999, pp. 121-128.
- [2] D. Q. Nguyen, R. Fedkiw, and H. W. Jensen, "Physically based modeling and animation of fire," Proc. ACM SIGGRAPH 2002, pp. 721-728 (2002)
- [3] B. Kang, Y. Jang, and I. Ihm, "Animation of chemically reactive fluids using a hybrid simulation method," Proc. Eurographics 2007, pp. 199-208 (2007).
- [4] Z. Yuan, F. Chen, and Y. Zhao, "Pattern-guided smoke animation with lagrangian coherent structure," ACM Transaction on Graphics, Article No.136, pp. 30 (2011).
- [5] 佐藤 周平, 森田 拓也, 土橋 宜典, 山本 強, "基本速度場による流体アニメーションの高解像度化," 電子情報通信学会 2013, pp. 338-345 (2013).

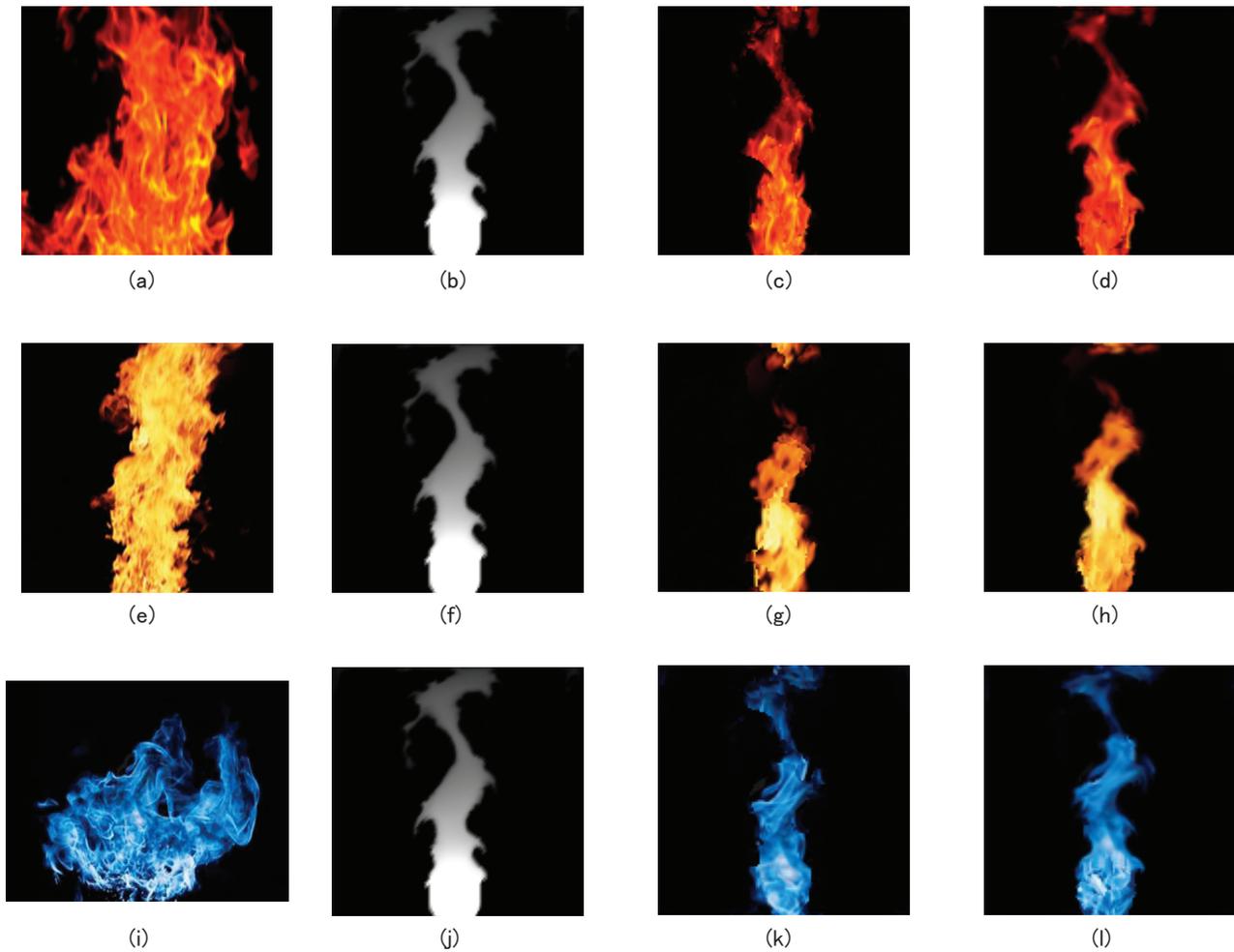


図 7 実験結果

- [6] T. Kim, N. Thurey, D. James and M. Gross, "Wavelet turbulence for fluid simulation," ACM Transaction on Graphics, Volume 27, 3 ,Article No.50 (2008).
- [7] Makoto Okabe, Ken Anjyo, Rikio Onai, "Creating Fluid Animation from a Single Image using Video Database," Proc. Pacific Graphics 2011.
- [8] M. Ashikhmin, "Synthesizing Natural Textures," Proc. the 2001 symposium on Interactive 3D graphics, pp. 217-226 (2001).
- [9] R. Fedkiw, J. Stam, and H. W. Jensen, "Visual simulation of smoke," Proc. ACM SIGGRAPH 2001, pp. 15-22 (2001)