Vol.2014-CG-157 No.22
Vol.2014-CVIM-194 No.22
2014/11/21

3 次元流体映像の変形と補間

谷 翼[1]　土橋 宜典[1,2]　佐藤 周平[3]　山本 強[1]

近年，コンピュータグラフィックス(CG)の進歩により，映画やゲーム，アニメなど様々な場面において，CG が多く用いられるようになっている．これら CG における映像生成の手段の一つとして，流体シミュレーションにより物理現象を再現する研究が広く行われている．しかし，写実的な映像を生成するためには，高精度なシミュレーションが必要となり，計算コストが膨大なものになってしまう．そこで本研究では，流体シミュレーションから得られた２つの密度場を変形及び合成することで，それらの中間となる形状を補間する手法を提案する．

Deformation and Interpolation of 3D fluid movie.

TSUBASA TANI[1]　YOSHINORI DOBASHI[1,2]
SYUHEI SATO[3]　TSUYOSHI YAMAMOTO[1]

1. はじめに

近年，映画やゲームなどにおいて，流体を含む様々な映像が CG で表現されるようになった．特に，流体などの自然現象は，物理方程式に基づいたシミュレーションを行うことにより，写実的な映像を生成することができる．しかし，流体解析を利用した物理シミュレーションは計算コストが非常に高く，映像のリアリティや複雑さを追求するほど計算時間が増大してしまう．また，所望の映像を生成するためには，一般に数多くのパラメータを試行錯誤的に決定し，シミュレーションが理想的な結果を出力するように調整するといった煩雑な作業を伴う．従って，所望の映像を得るまでには高コストのシミュレーションを繰り返し実行する必要があり,それには膨大な時間がかかってしまう．

そこで本研究では，流体シミュレーションから得られた複数の密度場から，その中間の流体形状を補間するための手法を提案する．これにより，限られたパターンの数のシミュレーション結果を用いて，ユーザは再度シミュレーションを行うこと無く，環境条件の変化に対応して形状も変化するような流体映像を低コストで生成することができる．提案手法では，2 つの密度場それぞれに対して，後述する移動最小二乗法を用いた変形処理を施し，2 つの入力密度場のおおよその形状を一致させた上で密度値の補間及び合成を行う．

2. 関連研究

Stam は，タイムステップを大きくとった場合でも，Navier-Stokes 方程式を安定的に解く方法を提案し，CG における実用的な流体シミュレーション手法を確立した[1]. Stam の手法以降，様々な流体現象を対象とした数多くの解析手法が提案されており，それらの手法の詳細は[2]にまとめられている．しかし，流体シミュレーションは計算コストが非常に高く，結果を生成するまでに多大な時間がかかってしまう．また，所望の流体の動きを作成するためにはシミュレーションのパラメータを試行錯誤的に調整しなければならない.

そのような流体シミュレーションの問題を解決するための手段として，事前に計算した流体速度場のデータベースを用いて実行時のシミュレーションを高速化する手法が提案されている[3][4]．しかし，これらの手法で様々な流れを表現するためには，膨大な前計算データとその作成のための計算時間が必要となる．また，上記のアプローチに基づき，高速に再シミュレーションを行う手法が提案されている[5]．この手法により，シミュレーションにより作成された流体の速度場に対し，そのパラメータを変えた際の流れ場を再度高速にシミュレーションする事ができる．しかしこの手法では，単一のデータを対象としており，流れの全体的な方向を変えるなど，大きな変更を行うことはできない. また, 佐藤らは 2 次元のシミュレーションについて，2 つの流体速度場の中間を極座標補間により算出することで，2 つの流れの中間の速度場を生成する手法を提案した[6]．この手法では，新たに補間した速度場を用いて移流を行うため，大域的な流れを変化させたアニメーションを生成させることができる．しかし，この手法は 3 次元のシミュレーションを対象としていない.

3. 提案手法の概要

本研究の目的は，3 次元シミュレーションによって生成された形状の異なる複数の密度場が存在する場合に，その中間形状を補間によって作ることである．形状の異なる密度場をただ単純に足しあわせても，当然ながら別々の形状

1　北海道大学
Hokkaido University
2　独立行政法人科学技術振興機構 CREST
JST CREST
3　UEI リサーチ
UEI Research

Vol.2014-CG-157 No.22
Vol.2014-CVIM-194 No.22
2014/11/21

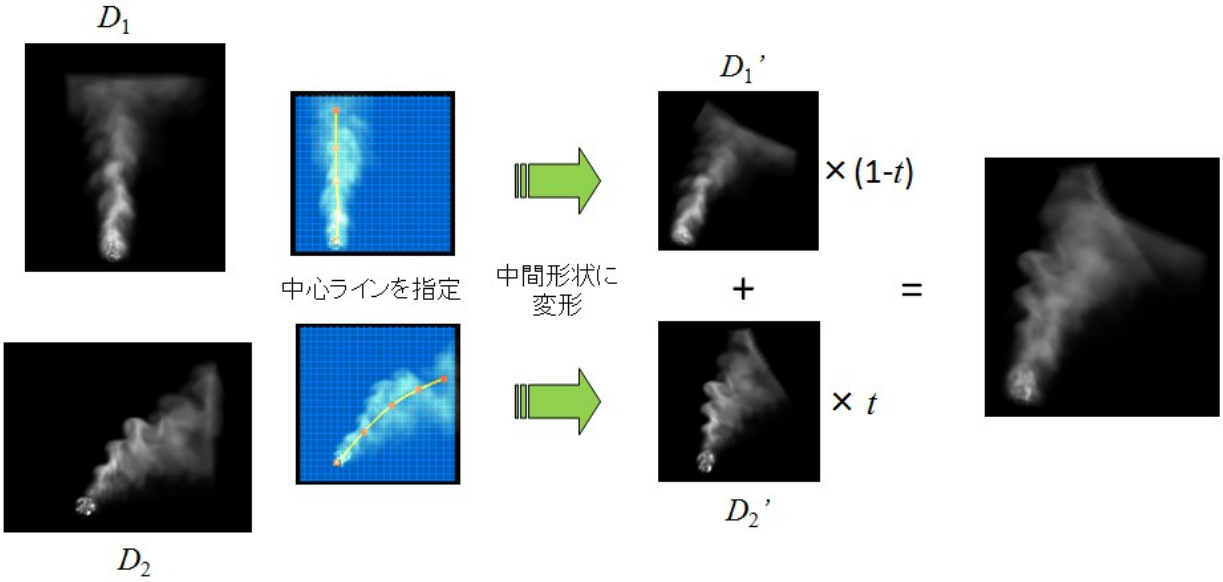


図 1 提案手法の概要

の流体が重なって表示されるだけであり，中間形状を生成できているとは言えない．また，単一の密度場に対して変形処理を行っただけでは，変形により元の形状との差異が大きくなるほど不自然な動きが目立つようになってしまう．そこで，提案手法では，入力となる複数の流体密度場の対応点を指定し，その対応点が一致するように変形した上で足し合わせる．これにより自然な中間形状を生成できる．

提案手法の概要を図 1 に示す．まず，3 次元流体シミュレーションにより複数の密度場を生成する(図 1 では二つ)．次に，ユーザにより対応の基準となる流体の中心ラインを指定する．そして，それぞれの中心ラインをユーザの指定した割合 t で補間し，中心ラインが一致するようそれぞれの密度場に対して変形処理を施す．変形した密度場を割合 t で合成することで，中間の密度場を生成する．割合 t を変化させることで，形状変化するアニメーションを作成することが可能である． 密度場の変形処理には，Schaefer らによる移動最小二乗法を用いた画像変形[7]の手法を用いる．この手法はアルゴリズムが単純であり，3 次元のシミュレーションにおいても容易に実装することができる．

以降，4 節において流体シミュレーション，5 節において Schaefer らの変形手法，6 節においてこれを用いた密度場の変形・補間方法について述べる．

4. 流体シミュレーション

本稿では，非圧縮性の流体シミュレーションから得られた流体の密度場を使用する．流体の動きは，次の Navier-Stokes 方程式(以下, NS 方程式)を解くことで計算される．

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{f} \qquad (1)$$

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

$\mathbf{u}$は流体の速度場，ρは流体の密度，pは圧力，νは動粘性係数，$\mathbf{f}$は重力や風などの外力である．式(1)は速度場の時間発展を表す方程式であり，右辺第 1 項から，それぞれ移流項，圧力項，拡散項，外力項と呼ばれる．また，式(2)は連続の式と呼ばれており，非圧縮性流体を扱う場合のみ成り立つ．また，Fedkiw らが提案した手法[8]を用いて，乱流を付加する．

次に，上記の方程式を用いて煙をシミュレーションする方法を説明する．NS 方程式により得られた速度場にしたがって煙の密度を以下の式によって移流させる．

$$\frac{\partial D}{\partial t} = -(\mathbf{u}\cdot\nabla)D + D_s \qquad (3)$$

Dは煙の密度，D_sは煙の発生源から追加される密度量を表す．提案手法は，これらの方程式を解くことで得られた密度場を変形・補間する．従って，NS 方程式や式(3)による移流の計算は，補間元となる入力密度場を生成する際にのみ必要となる．

5. Schaefer らによる画像変形手法

Shaefer らによる画像変形手法はアルゴリズムがシンプルで高速に実行することができるため，提案手法のように 3 次元のボリュームデータに適用しても高速に動作する．この手法では変形時に考慮する回転や拡大縮小等の要素から，アフィン変形，類似性変形，リジッド変形の 3 つの変形方法について述べている．提案手法においては，時間変化する流体に対して本手法を適用するため，空間上の拡大縮小を行うと，それに応じて流体の速度も一部分が不自然に変化してしまう．そこで提案手法では，変形において拡大縮小を許容せず，回転のみを考慮するリジッド変形を採用している．このリジッド変形を用いた画像変形手法について以下で説明する．

Vol.2014-CG-157 No.22
Vol.2014-CVIM-194 No.22
2014/11/21

前提条件として，ここでの二次元画像の変形は，画像を格子状のメッシュで分割し，メッシュ上の各頂点が移動することにより画像全体の変形が行われるものとする．このとき，元画像上にユーザが設定した制御点をp_i，変形後の各制御点の位置をq_iとおくと，メッシュ上の各頂点vを動かす関数$l_v(x)$は，次の式の値が最小になるように定義される．

$$\sum_i w_i \left| l_v(p_i) - q_i \right|^2 \qquad (4)$$

p_i及びq_iは二次元の行ベクトルで表される．また，重みw_iは，vからの距離が近い制御点ほど値が大きくなるように，次の式で定義される．

$$w_i = \frac{1}{\left| p_i - v \right|^{2\alpha}} \qquad (5)$$

αの値は基本的に 1 としてよい．これらの式から，線形代数に基づいた式変換により，メッシュ上の各頂点vをリジッド変形により動かす関数$f_r(v)$を求めると次の式(6)が得られる．なお，詳細な導出過程については論文[7]を参照して欲しい．

$$f_r(v) = |v - p_*| \frac{\sum_i [(q_i - q_*)A_i]}{|\sum_i [(q_i - q_*)A_i]|} + q_* \qquad (6)$$

ここで，p_*，q_*，A_iはそれぞれ次の式で与えられる．

$$p_* = \frac{\sum_i w_i p_i}{\sum_i w_i}$$

$$q_* = \frac{\sum_i w_i q_i}{\sum_i w_i}$$

$$A_i = w_i \begin{pmatrix} p_i - p_* \\ -(p_i - p_*)^\perp \end{pmatrix} \begin{pmatrix} v - p_* \\ -(v - p_*)^\perp \end{pmatrix}^T$$

$\perp$は直交するベクトルを意味する演算子であり，$(x, y)^\perp$は$(-y, x)$となる．ここで使われている変数は全て数値が与えられるため計算が可能である．特にA_iについては，制御点の位置p_iが与えられれば変形後の位置q_iに関わらず一定であるため，最初に一度計算すれば以降は更新する必要がない．

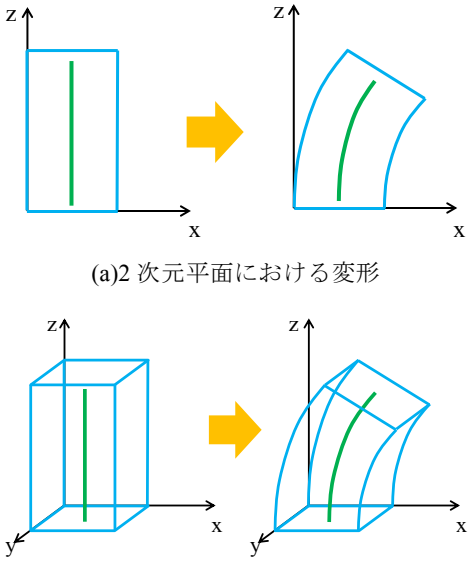
以上が，2 次元の画像に対して，制御点をベースにしたリジッド変形を施す際の計算処理である．提案法においてはユーザが指定したラインをベースに変形を行うが，これはライン上に細かく制御点を取り，ラインの変形に合わせて制御点も移動させることで実装している．

6. 密度場の変形と補間

4 節の方法により生成した異なる密度場に対し，5 節で述べた変形方法を適用して，中間の密度場を生成する方法を説明する．

6.1 中心ラインの指定と変形・補間処理

簡単のため，入力として二つの密度場 D_1 および D_2 を考える(図 1 参照)． 2 つの密度場それぞれにおいて，その形状の中心となるラインをユーザが指定する．提案手法における密度場の変形処理は，ここで指定した中心ラインを制御の基準として行う．そして，変形後の中間形状の流体における中心となるラインを，生成した 2 つの流体の中心ラインから線形補間により求める．ここで，中間の状態を表す変数 t ($0 \leqq t \leqq 1$)を導入する．$t = 0$ のときはD_1，$t = 1$ のときはD_2の形状と完全に一致し，$t = 0.5$ の時は 2 つの生成密度場のちょうど中間の形状を表す．本稿におけるラインを基準とした変形処理は，変形前後の基準となるラインの上に細かく制御点を配置することで行っている．生成した 2 つの密度場において指定した中心ライン上の制御点を，それぞれp_i，q_i（i=1,2,…,n）とおく．p_iは密度場 D_1に，q_iは密度場 D_2にそれぞれ対応する．変形後の中間形状における中心ラインは，次の式によって求められる点r_iの集合によって表される．



(a)2 次元平面における変形

(b)奥行き方向への変形処理の適用

図 2： 3 次元空間における変形処理

$$r_i = (1 - t)p_i + tq_i \qquad (7)$$

このようにして求められた中間形状の中心ラインと，生成した密度場に対して指定した中心ラインが一致するように，2 つの生成密度場に対して変形処理を行う．

密度場の変形と補間処理について説明する．変形については，基本的な処理は前節で説明した Schaefer らによる画像変形手法と同様である．まず視点方向から見た 2 次元平面に対して，ラインベースの変形処理を行う．続いてその変形を視点から見た奥行き方向の全ての面に対して適用することで，3 次元空間全体の変形を行う（図 2）．そのようにして得られたD_1，D_2の変形後の密度場をそれぞれD_1'，D_2'とおく．最終的な補間結果となる中間形状の密度場D_cは t を用いて次の式で合成される．

$$D_c = (1 - t)D_1' + tD_2' \qquad (8)$$

6.2 速度の調整

前節で述べた方法により，変形により形状を一致させた上で密度を補間することで，自然な合成を行うことができ

る．しかし，二つの流れの速度が異なる場合，合成した際にもその違いが残ってしまい，合成結果に不自然さが生じてしまう．そのため，形状だけでなく，流れの速さについても調整する必要がある．そこで，シミュレーションの段階において，それぞれの流体のおおよその中心となる格子点をユーザが指定し，その格子点における速さの平均値も記録する．そして，記録した速さが一致するように，一方の流体の再生速度を調整する．例えば，2 つの流体の速さをそれぞれv_1，v_2とおき，v_1をv_2に合わせる場合を考えると，D_1の再生速度は次の式で表されるp_1倍になる．

$$p_1 = \frac{v_2}{v_1}$$

速さの調整により入力密度場から小数点フレームを描画する必要が生じた場合は，その前後の整数フレームから密度値を線形的に補間する．

7. 実験結果・考察

提案手法により，2 つの煙の密度場からその中間形状を生成した結果を図 3 に示す．実験環境は，CPU: Intel Core i7-3770 CPU，メモリ：3.5GB，グラフィックボード: NVIDIA GeForce GTX 660 である．

まず，提案法の有効性を示すため，流体シミュレーションによって変形を行った場合との比較を図 3 に示す. 図 3(a)は，流体シミュレーションによる結果であり，左から外力として風を吹かせている．風の強さを徐々に変化させて煙が変形していく様子を示している．図 3(b)は，提案法によって，無風状態でのシミュレーション結果と風が強い状態でのシミュレーション結果から補間して生成したものである．図 3(b)の上端および下端がシミュレーションによって得られた煙に対応し，その間が補間によって生成された結果である. シミュレーションの解像度は 100x50x100 である. この図から提案法により，自然な形状変化が実現できていることがわかる．図 3(a)では，180 タイムステップのシミュレーションを行っており，計算時間は 174 秒であった．図 3(b)の場合，2 つのシミュレーションデータを生成するために，合わせて 367 秒の時間が必要になるが，変形・補間処理は 40 秒でよく, 高速に映像生成を行うことができる.

次に，提案法を蒸気機関車から発生する煙のシミュレーションに適用した例を図 4 に示す．蒸気機関車の走行速度に応じて煙の変形・補間を行った．密度場の解像度は，200x100x200 である．シミュレーションによるデータ生成には 40 分の時間を要した．変形・補間処理による中間形状の生成は 6 分であった．

8. まとめ

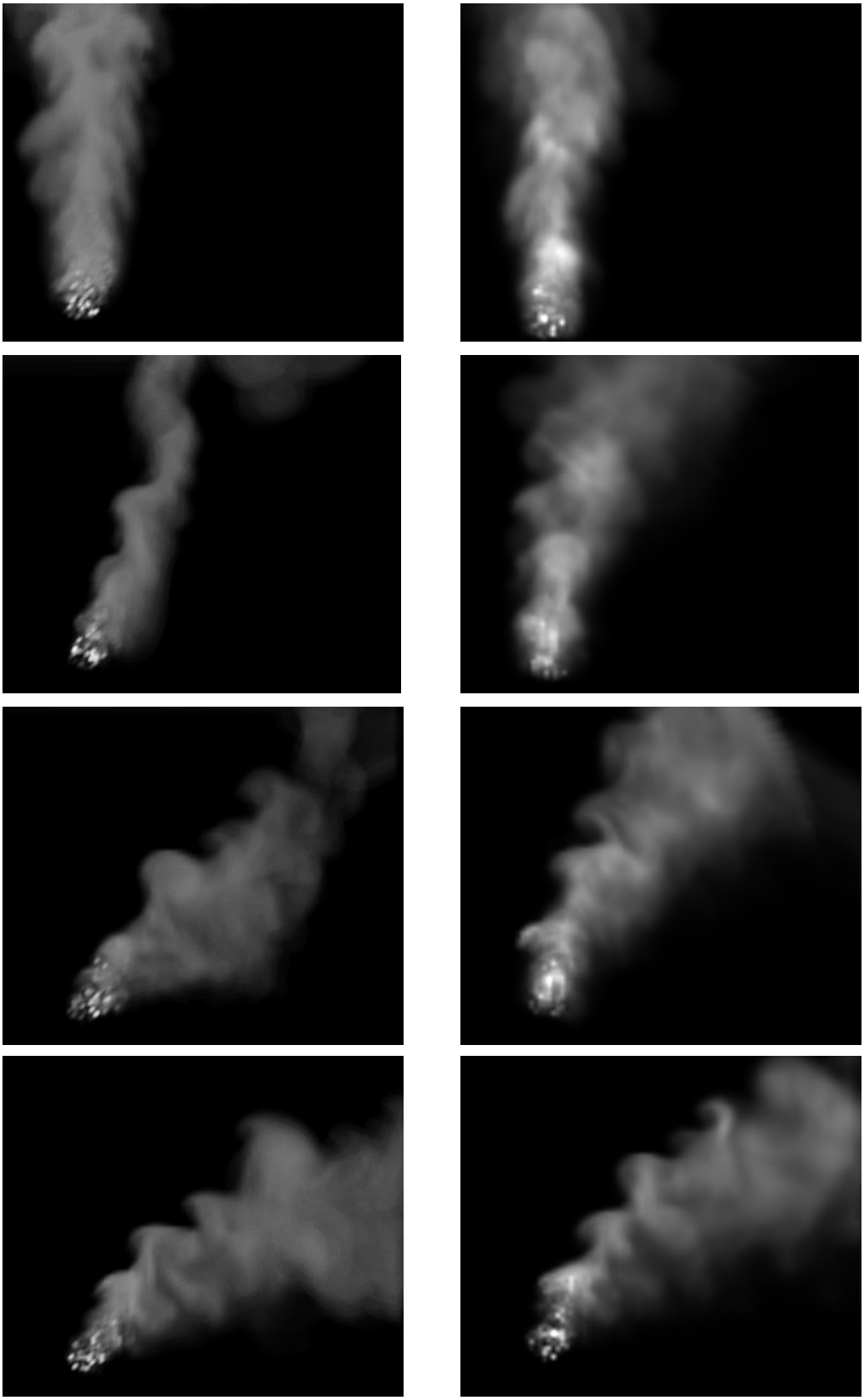
本稿では，2 つの流体の密度場に対して，移動最小二乗法を用いた変形処理を施した上で合成することで，2 つの流体の中間の形状を補間する手法を提案した．提案手法により，3 次元シミュレーションによって生成された形状の異なる複数の密度場から，その中間形状を新たにシミュレーションを行うことなく作成することが可能となった．

今後の課題としては，まず複雑な形状への変形にも対応することが挙げられる．現在の変形処理は 2 次元平面での変形をそのまま奥行方向へ適用するため，例えば螺旋状に変形するといったような 3 次元的な変形には対応していない．また，生成密度場の太さの違いについては考慮していないため，太さが大きく違う流体間の補間に対応できないという問題があり，この点についても今後解決の必要がある．

参考文献

[1] J. Stam : Stable fluids, In *Proceedings of ACM SIGGRAPH 1999, Annual Conference Series*, 121-128(1999).
[2] R. Bridson : *Fluid Simulation for Computer Graphics*, AK Peters (2008).
[3] A. Treuille and A. lewis and Z. Popovic : Model reduction for real-time fluids, *ACM Transactions on Graphics 25*, 3, 826-834 (2006).
[4] M. Wicke and M. Stanton and A. Treuille : Modular bases for fluid dynamics, *ACM Transactions on Graphics 25*, 3, 826-834 (2006).
[5] T. Kim, and J. Delaney : Subspace fluid re-simulation, *ACM Trransactions on Graphics 32*, 4, Article 62 (2013).
[6] 佐藤 周平, 土橋 宜典, 山本 強, 西田 友是 : 極座標変換による異なる流体流れ場の補間手法, 情報処理学会 グラフィクスと CAD 研究会 第 155 回研究発表会 (2014).
[7] S. Schaefer, T. McPhail, J. Warren : Image Deformation Using Moving Least Squares, In *Proc. SIGGRAPH 2006*, 533-540 (2006).
[8] R. Fedkiw, J.stam, and]H.W. Jensen : Visual simulation of smoke, In *Proc. SIGGRAPH 2001*, 15-22 (2001).

Vol.2014-CG-157 No.22
Vol.2014-CVIM-194 No.22
2014/11/21



(a) (b)

図 3 流体シミュレーションと提案法の比較．(a) 風によって変形する煙を流体シミュレーションによって生成した結果．(b)提案法によって二つのシミュレーション結果から生成した結果.

Vol.2014-CG-157 No.22
Vol.2014-CVIM-194 No.22
2014/11/21

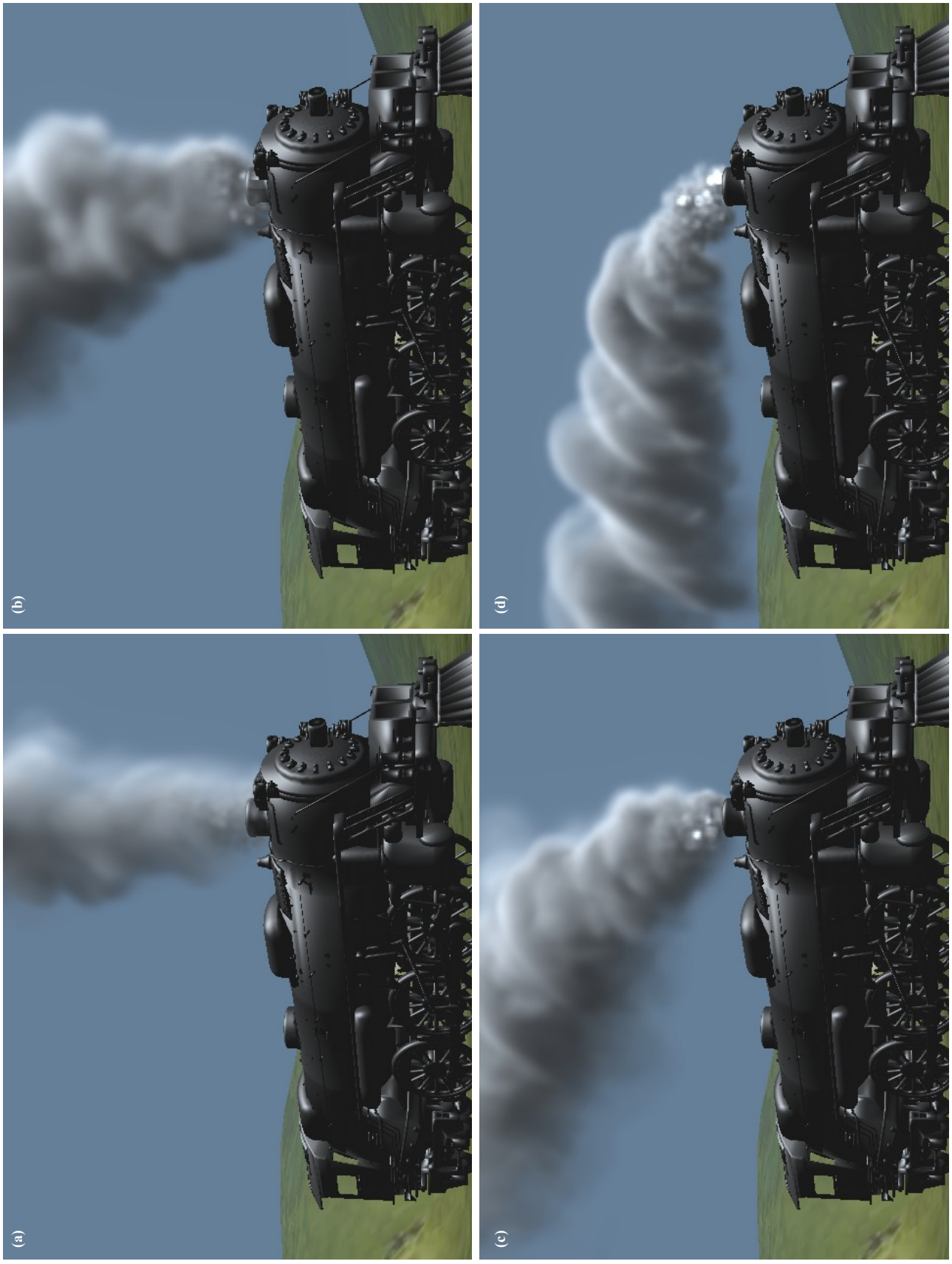


図4 蒸気機関車から立ち上る煙の例.