Vol.2010-CG-138 No.14
2010/2/12

粒子ベース流体シミュレーションを用いた炎のリアルタイムレンダリング

稲垣 智[†1] 井村誠孝[†2] 池田 聖[†1]
眞鍋佳嗣[†1] 千原國宏[†1]

映画やゲームにおいては，炎を写実的かつリアルタイムにレンダリングする手法が求められている．炎の CG はエンタテインメントのみならず，防災のための火災シミュレータなどにおいても有用である．本論文では，炎を粒子ベース流体シミュレーションに基づき，写実的かつリアルタイムにレンダリングする手法を提案する．炎を写実的にレンダリングするには，三つの条件を満たすことが重要である．一つ目の条件は，炎が風のような力の影響を受けて変形することである．二つ目の条件は炎の明るさと色が正確であることである．三つ目の条件は，炎の周辺に陽炎が発生することである．提案手法はこれら三つの条件を満たすように粒子ベース流体シミュレーションすることで，炎を写実的かつリアルタイムにレンダリングする．提案手法を実験し，写実的かつリアルタイムにレンダリングできることを確認した．

Real-time Rendering of Flame Using Particle-based Fluid Simulation

Satoru Inagaki,[†1] Masataka Imura,[†2] Sei Ikeda,[†1]
Yoshitsugu Manabe[†1] and Kunihiro Chihara [†1]

Photo-realistic real-time rendering of flame is required in movie or video game. In particular, rendering of flame is useful for not only entertainment but also fire simulation for firefighting training. In this paper, we present a method for simulating the behavior of flame and rendering it as photo-realistic CG in real-time. It is important to meet three requirement for photo-realistic and real-time rendering of flame. First requirement is that flame become deformed by force such as wind. Second requirement is that brightness and color of flame is true. Third requirement is that air turbulence arises around flame. Our method photo-realistic and real-time renders flame by particle-based fluid simulation meeting three requirement. We make sure that our method photo-realistic and real-time render flame by experiment.

1. 序　　論

映像制作において炎は撮影が危険であり，頻繁に CG で生成される．また，炎の CG はゲームにおいて画面を写実的にレンダリングするために利用される．さらに，炎の CG はエンタテインメントのみならず防災のための火災シミュレータなどにおいても有用である．炎は形状が不定であり，物理ベースシミュレーションを用いた手法では流体として扱われる．近年，ハードウェアの進化により，リアルタイムで流体シミュレーションが可能になった．また，グラフィックスハードウェアのプログラミングが可能になり，レンダリング結果を詳細に制御することが可能になった．本論文では，炎の挙動を流体シミュレーションに基づき，写実的かつリアルタイムにレンダリングする手法を提案する．

炎を写実的にレンダリングするには，次の三つの条件を満たすことが重要である．

- 炎は風のような力の影響を受けて変形すること
- 炎の明るさと色が正確であること
- 炎の周辺に陽炎が発生すること

提案手法では，これら三つの条件を満たすように炎を流体シミュレーションし，写実的にレンダリングする．

2. 関 連 研 究

炎を写実的にレンダリングするには，流体シミュレーションを用いた手法が最も有効である．流体シミュレーションは流体の定義により格子法と粒子法に分類できる．

2.1 格 子 法

格子法では流体が流入する可能性のある空間をその空間に固定された立方体の格子に分割し，流体の密度や速度などの物理量を各格子点で定義する．格子法では各格子点で定義した物理量の時間変化を計算することで流体の挙動をシミュレーションする．格子法は空間を立方体の格子に分割するため，勾配のような空間微分の計算が容易かつ正確であるという利点がある．しかし，格子法では流体が流入する可能性のある空間は，初期の時点で流体が

†1 奈良先端科学技術大学院大学
NAIST
†2 大阪大学
Osaka University

存在せずとも，その後のシミュレーションで流入する可能性があるならば，格子に分割しシミュレーションしなければならないという欠点がある．また，格子法で流体を精密にシミュレーションするには，空間を大量の格子に分割する必要があり，格子数に応じた計算時間を必要とする．例えば，Fedkiw らは格子法で炎の挙動をシミュレーションする手法を提案しているが[1)2)]，Fedkiw らの手法は空間を大量の格子に分割しているため，正確ではあるが 1 フレームのレンダリングに約 3 分以上の計算時間を必要としている．従って，リアルタイムレンダリングには適さない．

2.2 粒 子 法

粒子法は流体を粒子の集合で定義する．ここで粒子とは分子や原子ではなく流体の小さな塊であるといえる．例えば流体を水とすると，粒子は水滴であるといえる．提案手法では炎の小さな塊であるといえる．粒子は流体が持つ速度や温度などの物理量を定義するための一定の体積・質量を保持している．

粒子法は粒子が持つ物理量の時間変化を計算する．粒子は任意の位置へ移動するため，空間微分の計算式が複雑であるという欠点がある．しかし，粒子法では流体自体をシミュレーションするので，流体が存在しない空間をシミュレーションする必要がなく，計算が高速であるという利点がある．従って，リアルタイムレンダリングに適している．

竹下らは炎と空気を粒子法でシミュレーションし，炎の CG をレンダリングする手法を提案しているが[4)]，竹下らの手法はリアルタイムレンダリングを目的としたものではない．

3. 提 案 手 法

提案手法はシミュレーションとレンダリングに分けることができる．シミュレーションでは炎の挙動を粒子法で流体シミュレーションする．レンダリングではシミュレーション結果をレンダリングする．

4. シミュレーション

炎の実態は燃焼中の気体燃料である．気体は流体であり粘性の影響が小さいので，流体シミュレーションでは粘性が無い流体の運動を表すオイラー方程式に基づいて計算を行う．オイラー方程式を計算することで，力が加わることによる流体の運動変化を求める．コンピュータグラフィックスのための流体シミュレーションは”Fluid simulation”[3)] に詳細が書かれている．

4.1 オイラー方程式

式 (1) はオイラー方程式である．

$$\rho \mathbf{a} = -\nabla p + \mathbf{f} \tag{1}$$

ここで，$\mathbf{a} = (a_x, a_y, a_z)$ は流体の加速度ベクトルである．ρ は流体の密度，p は圧力を表している．圧力は流体が加える単位体積あたりの力を表している．$\mathbf{f}$ は外力を表している．外力は流体の表面ではなく，流体全体に加わる単位体積あたりの力を表している．

粒子を用いて流体をシミュレーションするとき，提案手法で各粒子は炎の小さな塊を表しているといえる．各粒子は次の 4 つの物理量を保持している．

- 質量 m
- 速度ベクトル $\mathbf{u}$
- 位置ベクトル $\mathbf{x}$
- 温度 T

質量は時間変化しないため定数である．温度はシミュレーションを通して定数として扱う．位置ベクトルと速度ベクトルの時間変化は，オイラー方程式により計算する．各粒子の位置ベクトルと速度ベクトルを計算するには，各粒子に加わる力を計算する必要がある．

4.2 SPH(Smoothed Particle Hydrodynamics) 法

SPH 法は空間中の離散的な位置に存在する粒子の物理量を用いて，滑らかで連続な物理量の場を定義し，定義した物理量の場の時間変化を流体の方程式を用いて計算する手法である．各粒子でオイラー方程式を計算するには，空間中の離散的な位置にある各粒子で与えられた物理量のみを用いて，滑らかで連続な物理量の場を定義しなければならない．SPH 法は滑らかで離散的にサンプリングされた物理量の場を定義する．これは滑らかな核関数 $W(r)$ を用いて定義される．核関数 $W(|\mathbf{x} - \mathbf{x}_i|)$ は，粒子 i の位置 $\mathbf{x}_i$ の付近でスカラーを重み付けする関数である．一般的に，核関数は poly6 核関数が用いられる．なぜなら，poly6 核関数には計算時間がかかる平方根が含まれないからである．

$$W_{poly6}(r) = \frac{315}{64\pi d^9} \begin{cases} (d^2 - r^2)^3 & (0 \le r \le d) \\ 0 & otherwise \end{cases} \tag{2}$$

ここで，r は粒子間距離，d は粒子の中心で定義した物理量を平滑化する範囲である．式 (2) は，$|r|$ が減少するほど値が大きくなる左右対称の関数である．横軸に r，縦軸に $W(r)$ を取ったグラフを書くと，山のような形をしている．核関数を用いると，各粒子の質量と位置から滑らかな密度場を計算できる．任意の物理量は密度を基準に計算するので，シミュ

レーションの一番最初に密度を計算する必要がある．式 (3) は密度場を定義する．

$$\rho(\mathbf{x}) = \sum_j m_j W(|\mathbf{x} - \mathbf{x}_j|) \tag{3}$$

式 (3) は各粒子の質量を核関数 W で重み付けし，合計することで密度場を定義している．核関数 W の値は粒子 j の位置 $\mathbf{x}_j$ から遠ざかるほど小さくなる．従って，密度は粒子 j に近づく程増加し，離れるほど減少する．また，粒子 j の質量が大きいと，密度も大きくなる．粒子 i の位置における密度は $\rho_i = \rho(\mathbf{x}_i)$ と省略して書く．核関数が $\int W(|\mathbf{x} - \mathbf{x}_i|)d\mathbf{x} = 1$ を満たすように正規化されている場合，密度場を空間で積分すると質量の合計と一致する．

$$\int \rho(\mathbf{x})d\mathbf{x} = \sum_j \left(m_j \int W(|\mathbf{x} - \mathbf{x}_j|)d\mathbf{x} \right) \tag{4}$$

$$= \sum_j m_j \tag{5}$$

従って，核関数としては正規化されたものを用いる．各粒子の位置における密度を用いると，各粒子の位置において定義した任意の物理量 A_i から物理量の場 A_s を計算できる．

$$A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(|\mathbf{x} - \mathbf{x}_j|) \tag{6}$$

式 (6) は，密度と共に変化する任意の物理量の場を定義する．式 (6) の両辺の勾配を求めると，m_j, A_j, ρ_j は定数なので式 (7) となる．

$$\nabla A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(|\mathbf{x} - \mathbf{x}_j|) \tag{7}$$

式 (7) は物理量の勾配の計算が，核関数の勾配の計算のみで行えることを表している．式 (3) を用いると，各粒子の質量から任意の位置における密度を計算できる．各粒子の位置における密度を計算することで，各粒子の密度を計算できる．式 (6) を用いると，各粒子の質量・密度・任意の物理量から，任意の位置における物理量を計算できる．各粒子の位置における物理量を計算することで，各粒子の物理量を計算できる．式 (7) を用いると，各粒子の質量・密度・任意の物理量から，物理量の勾配を計算できる．

4.2.1 SPH 法によるオイラー方程式の計算

式 (7) を用いると，任意の物理量の勾配を任意の位置で計算できる．従って，オイラー方程式の右辺に現れる勾配を各粒子の位置で計算できる．ここで，説明のためにオイラー方程式の表記を変更する．式 (1) の右辺の圧力を $\mathbf{f}^{\text{pressure}} = -\nabla p$ と置き，外力を $\mathbf{f}^{\text{external}} = \mathbf{f}$ と置く．そして，$\mathbf{f} = \mathbf{f}^{\text{pressure}} + \mathbf{f}^{\text{external}}$ と置く．すると，オイラー方程式は $\rho\mathbf{a} = \mathbf{f}$ と書ける．粒子 i の位置における力を $\mathbf{f}_i$，密度を ρ_i，加速度を $\mathbf{a}_i$ と書くと，各粒子の位置においてオイラー方程式は $\rho_i\mathbf{a}_i = \mathbf{f}_i$ と書ける．両辺を ρ_i で割ると，粒子 i の加速度は式 (8) となる．

$$\mathbf{a}_i = \frac{\mathbf{f}_i}{\rho_i} \tag{8}$$

4.2.2 圧　　力

式 (7) を適用し，圧力勾配を計算すると式 (10) となる．

$$\mathbf{f}_i^{\text{pressure}} = -\nabla p_i \tag{9}$$

$$= -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|) \tag{10}$$

ここで，$p_i = p(\mathbf{x}_i)$ と省略して書いた．圧力 p は密度を元にした理想気体の状態方程式で計算される．

$$p_i = k(\rho_i - \rho_0) \tag{11}$$

ここで，k は温度に依存する気体定数であり，ρ_0 は密度の基準である．式 (3) を用いて各粒子の密度を計算すると，圧力 p を計算できる．ρ_0 を調節すると，圧力の基準を調節でき，k を調節すると圧力の大きさを調節できる．

4.2.3 外　　力

提案手法では外力として風力を考慮する．

4.2.4 式の対称化

式 (10) は式が対称でないため，対称化する．式 (12) は対称化した式である．

$$\mathbf{f}_i^{\text{pressure}} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|) \tag{12}$$

加速度 $\mathbf{a}_i = \mathbf{f}/\rho_i$ を考慮すると，式 (12) は対称である．式 (12) を用いて圧力勾配を計算する．

4.3 実　　装

フレームを更新するごとに以下の処理を行う．

(1) 粒子の追加
(2) 近傍粒子の探索
(3) 密度の計算
(4) 圧力勾配と外力の計算

(5)　速度と位置の計算

粒子の追加では，各粒子の位置ベクトルと速度ベクトルの初期値を与え，粒子を一定数追加する．位置ベクトルの初期値には，初期に粒子が現れる位置を与え，速度ベクトルの初期値には，粒子の初速を与える．位置ベクトルの初期値は燃焼の開始位置である．質量と温度には定数を与える．近傍粒子の探索では，密度と圧力勾配を計算するために，粒子間距離 r が有効範囲 d より小さい粒子の組み合わせを記憶する．密度の計算では，式 (3) を用いて，各粒子の密度を計算する．圧力勾配と外力の計算では，計算した密度を式 (11) に代入することで圧力を計算する．その後，式 (12) を用いて圧力勾配を計算する．さらに，外力を加える．外力には任意の値を代入できる．提案手法では風力をそのまま代入する．速度と位置の計算では，密度，圧力勾配，外力を式 (8) に代入することで，加速度ベクトルを計算する．最後に，加速度ベクトルから，速度ベクトルと位置ベクトルを計算する．

5. レンダリング

本節では，シミュレーション結果に基づいて，炎と陽炎をレンダリングする手法について述べる．

5.1 黒体放射

通常，物体は光を反射または屈折させる．例えば，鏡は光を反射させ，ガラスは光を屈折させる．それに対して，黒体は光を全て吸収すると仮定した物体である．黒体に入射した光は屈折も反射もしない．黒体に近い物体として炭が挙げられる．黒体放射は黒体からの光の放射である．提案手法では炎を黒体として扱い，放射光のみをレンダリングする．黒体放射は温度に依存する．例えば，低温の黒体は黒く見える．黒体放射はプランクが考案したプランクの式で表される．式 (13) はプランクの式である．

$$I(\lambda, T) = \frac{2\pi hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} \tag{13}$$

ここで，I は黒体からの放射光である．λ は波長 (nm)，T は温度 (K)，h はプランク定数，k はボルツマン定数，c は光速である．プランクの式は温度と波長の関数とみなせる．プランクの式に温度を代入すると，波長の関数となる．光の波長は色を決定する．色をディスプレイに表示するには，色を Red, Green, Blue で表す RGB 表色系に変換しなければならない．色を RGB 表色系に変換するには，各表色系の基礎となっている XYZ 表色系に変換する必要がある．XYZ 表色系の X，Y，Z はそれぞれ R，G，B に近い色を表している．放射光を XYZ 表色系で表すには，式 (14)(15)(16) を用いる．

図 1　温度と色の関係
Fig. 1　Relation between Temperature and Color

$$X = \frac{1}{c} \int_{360}^{800} I(\lambda, T) x(\lambda) d\lambda \tag{14}$$

$$Y = \frac{1}{c} \int_{360}^{800} I(\lambda, T) y(\lambda) d\lambda \tag{15}$$

$$Z = \frac{1}{c} \int_{360}^{800} I(\lambda, T) z(\lambda) d\lambda \tag{16}$$

ここで，$c = \int_{360}^{800} y(\lambda) d\lambda$ である．x，y，z はそれぞれ CIE（国際照明委員会）が定めた XYZ 等色関数を表している．可視領域は 360nm から 800nm なので，この領域で積分する．XYZ 表色系から RGB 表色系への変換は式 (17) を用いる．

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{17}$$

プランクの式から RGB 表色系への変換はリアルタイムで計算できない．そこで，プランクの式を事前に計算する．レンダリングに必要な温度の範囲でプランクの式を計算し，RGB 表色系に変換した結果をテクスチャとして保存する．図 1 は保存したテクスチャの一例である．テクスチャは温度と色の関係を表している．図 1 のテクスチャの温度の範囲は 1K から 10000K である．テクスチャはレンダリング時に温度で参照する．

5.2 炎のレンダリング

各粒子の温度を用いて図 1 に示すテクスチャの色を参照し，図 2 に示すテクスチャと乗算し，各粒子に貼り付けてレンダリングする．重なった部分は加算合成する．アルファ値は各粒子の密度に応じて決定する．図 2 のテクスチャは (r, g, b) のそれぞれで 8 ビットを使用しており，(r, g, b) のそれぞれが 0 から 255 の値を取る．図 2 のテクスチャは，poly6 核関数に座標を代入し，結果が 0 から 255 の範囲に含まれるように正規化することで作成する．

Vol.2010-CG-138 No.14
2010/2/12

図 2 粒子の密度分布
Fig. 2 Density of Particle

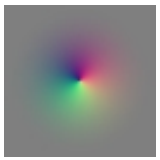

図 3 屈折の方向
Fig. 3 Direction of Refraction

5.3 陽　　炎

炎により熱せられた空気は膨張し密度が減少する．その結果，炎の周辺には密度差が発生する．光は密度が増加する方向へ屈折するため，炎の周辺は密度差による屈折で背景がゆがんで見える．

図 3 に示すテクスチャは各粒子の付近で光が屈折する方向を色で表している．すなわち，(r, g, b) = (x, y, z) である．図 3 のテクスチャは poly6 核関数の微分を計算し，計算結果が 0 から 255 の範囲に含まれるように正規化することで作成する．

各粒子に図 3 に示すテクスチャを貼り付けてレンダリングし，コンピュータグラフィックスにおける視点から見た光の屈折方向を表した画像を生成する．レンダリングの際，粒子が重なった部分は加算合成する．図 4 は視点から見た炎の周辺における光の屈折方向を表している．

図 4 を元に背景をゆがめてレンダリングする．すると，図 6 のように背景がゆがんでレンダリングされる．

6. 実　　験

実験用のプログラムを作成し，提案手法の有効性を確認した．風力にはマウスドラッグのベクトルをそのまま代入した．

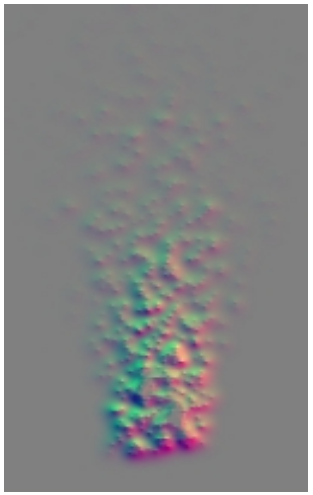

図 4 視点から見た屈折の方向
Fig. 4 Direction of Refraction from View Point

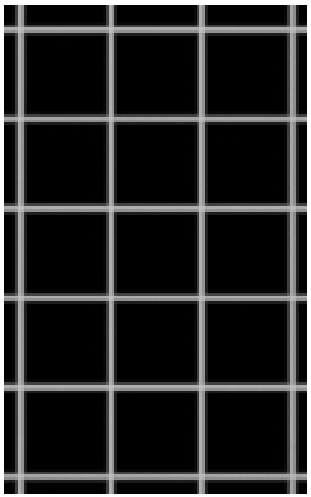

図 5 背景（陽炎無し）
Fig. 5 Background（without Heat Shimmer）

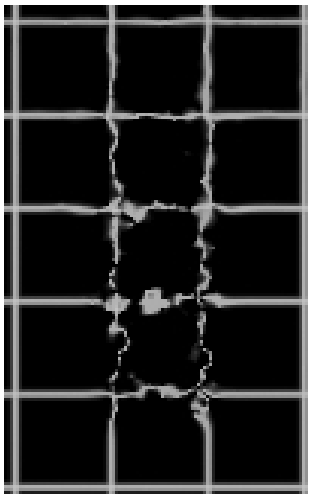

図 6 背景（陽炎有り）
Fig. 6 Background（with Heat Shimmer）

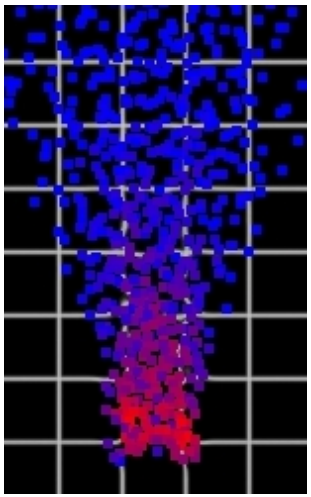

図 7 密度分布
Fig. 7 Density

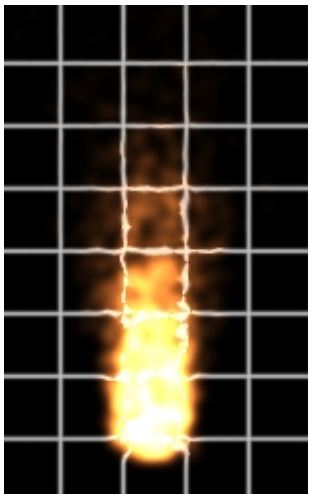

図 8 レンダリング結果
Fig. 8 Result

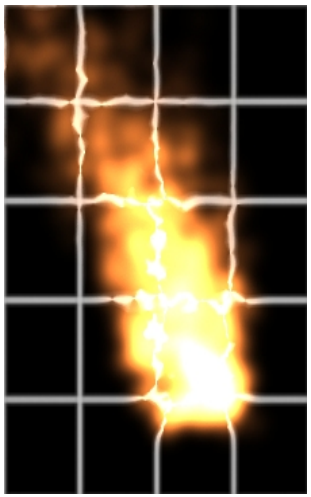

図 9 左方向へ力を加えた結果
Fig. 9 Result adding Left Force

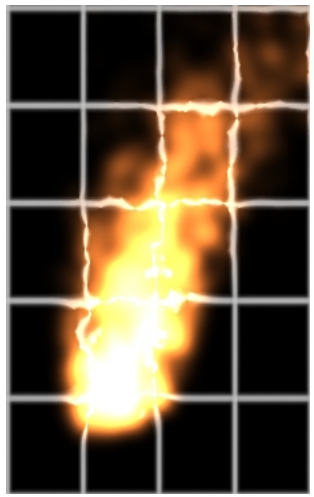

図 10 右方向へ力を加えた結果
Fig. 10 Result adding Right Force

6.1 実験環境

CPU は Core 2 Duo 2.13GHz を，GPU は NVIDIA GeForce 7600 GS を使用した．コンパイラは Microsoft Visual Studio .NET 2005 を使用した．言語は C++を，グラフィックスライブラリは OpenGL を，シェーダ言語は Cg 言語を使用した．

6.2 結　　果

図 7 は各粒子を正方形としてレンダリングした結果を表している．各粒子の色は赤いほど密度が高いことを表している．図 8 は最終的なレンダリング結果を表している．

図 9 はマウス操作により左方向に風力を与えた結果を，図 10 は右方向に風力を与えた結果を表している．いずれの結果も粒子数は約 3000 でフレームレートは 60fps となっている．

7. 考　　察

下部に追加された粒子は初速により上昇している．追加された直後の密度は高いので，圧力が上昇し，分散している．密度に応じてアルファ値を決定し，温度で色を決定しているので，炎の明るさと色を正確にレンダリングできている．また，風の影響を受けて変形している．さらに，陽炎で背景がゆがんでレンダリングされている．

8. 結　　論

本論文では，炎を写実的かつリアルタイムにレンダリングする手法を提案した．提案手法は 3 つの条件を満たす粒子ベース流体シミュレーションに基づき，炎を写実的にレンダリングする．実験によって，提案手法が有効であることを確認した．

参 考 文 献

1) Hong, J.-M., Shinar, T. and Fedkiw, R.: Wrinkled flames and cellular patterns, *ACM Trans. Graph.*, Vol.26, No.3, p.47 (2007).
2) Nguyen, D. Q., Fedkiw, R. and Jensen, H. W.: Physically based modeling and animation of fire, *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM, pp.721–728 (2002).
3) Robert, B., M.-F.M.: Fluid Simulation, *SIGGRAPH 2007 Course Notes* (2007).
4) Takeshita, D., Ota, S., Tamura, M., Fujimoto, T., Muraoka, K. and Chiba, N.: Particle-Based Visual Simulation of Explosive Flames, *IPSJ SIG Notes*, Vol.2002, No.77, pp.121–126 (2002).