

## 粒子ベース火炎レンダリング

間 淵 聡<sup>†1</sup> 藤 代 一 成<sup>†1</sup> 大 野 義 夫<sup>†1</sup>

本稿では粒子ベースによる火炎レンダリング法を提案する。可視化分野で考案された粒子ベースボリュームレンダリングと呼ばれる手法をベースとし、これを火炎の特性に合わせて合理的に改良した。これにより、従来の粒子ベースレンダリングでは困難であった、ディテールの表現とユーザの制御性を大きく向上させることができた。また、既存 API を活用することで、実装が容易かつ実行速度は高速であり、十分なインタラクティブ性を提供できる。さらに、本手法の要求する粒子データは座標と温度のみであるため、特定の粒子シミュレータに依存することなく、汎用的に用いることができる。

## Particle-Based Flame Rendering

SATOSHI MABUCHI,<sup>†1</sup> ISSEI FUJISHIRO<sup>†1</sup>  
and YOSHIO OHNO<sup>†1</sup>

In this article, we propose a particle-based flame rendering method. In the field of visualization, particle-based volume rendering has been introduced for the purpose of previewing. Our method is a reasonably-simplified version of the particle-based volume rendering. Compared with previous flame rendering methods, both detail representation and controllability are sufficiently improved. In addition to this, it is easily implemented using standard APIs, and its fast enough rendering speed allows us to adjust environmental settings on the fly. Moreover, our method requires only the position and temperature as particle attributes, and thus it would be used in conjunction with existing particle simulators.

<sup>†1</sup> 慶應義塾大学  
Keio University

### 1. 序 論

CGにおけるレンダリングは、固体、液体などのさまざまな現象でインタラクティブ性と写実性を提供できるようになりつつある。しかし火炎のレンダリングにおいては、近年でもなおインタラクティブ性のある手続き的手法が確立されていない。

格子ベースで定義された火炎データに対してはレイマーチングを用いた手法が確立されているが、これは計算コストが高い。粒子ベースで定義された火炎データに対しては、視点奥行き方向にソートされたポイントスプライトが用いられてきた。ディテールを表現するためには大量の点群が必要であるが、この手法は大量の点群を扱うのには適していない。

一方、可視化を目的とした粒子ベースボリュームレンダリングと呼ばれる手法が提案された。これは格子から大量の点群を生成し、Zバッファ法によるレンダリングを繰り返すことで擬似的に半透明を表現する手法である。しかしこの手法は粒子生成コストが高く、粒子の事前生成が必要となる。本稿ではこの粒子ベースボリュームレンダリングを合理的に簡略化し、粒子の事前生成が不要な、火炎に特化したインタラクティブレンダリング技法を提案する。

本稿では、次節で関連研究を紹介した後、3節で関連研究を土台としたアプローチを説明する。4節で提案するレンダリングモデルを述べ、5節で効果的な調整方法を説明する。最後に、6節で実装結果を、7節で結論と課題を述べる。

### 2. 関連研究

火炎レンダリングの関連研究について述べた後、半透明物体表現一般の研究について触れる。

#### 2.1 火炎レンダリング

火炎のレンダリング手法は以前から、主に映画製作の過程で研究されてきた。CGにおける流体アニメーションを最初に発表したのはReeves<sup>8)</sup>である。この研究では、炎の生成から消滅までを単純化し、80万程度の粒子を用いてそれらしく見せることに成功している。また、Inakage<sup>3)</sup>は、小規模火炎に特化したモデルを提案した。

近年の火炎レンダリング技法も、同様に映画製作の過程で研究されてきた。Nguyenら<sup>6)</sup>はシミュレーションとレンダリングを理論的に構築した。この研究ではレンダリングに再帰的レイマーチングを用いている。粒子によるレンダリング技法として、ポイントスプライトをベースとする、百万単位の粒子を用いた手法<sup>4)</sup>もある。近年では、ラフな粒子シミュレー

ションを CPU 上で行い、詳細なシミュレーションとレンダリングを複数の GPU 上で行う手法が Horvath ら<sup>2)</sup>によって提案された。これらは計算コストと写実性の両立を図っているが、映画製作を目的としており、まだなお充分なインタラクティブ性をもたせることは難しい。

手続き的なインタラクティブ火炎レンダリング手法は少なかったが、井村らは、粒子ベースインタラクティブ火炎レンダリング<sup>12)</sup>を提案した。屈折を用いてリアリティの向上を実現しているが、ディテールの表現を目的としたものではない。

### 2.2 A-Buffer

半透明物体一般の表現として、A-Buffer<sup>1)</sup>が用いられてきた。これはピクセル単位でフラグメントのリストを保持し、それらをもとに Z ソートを行う手法であるが、アルゴリズムが複雑であることからオフラインレンダリングに用いられてきた。近年では GPU の進歩により、ステンシルバッファを活用することで、A-Buffer が GPU 上で実装されている<sup>5)</sup>。ただし A-Buffer は主にポリゴンデータの表現に用いられ、ボリュームデータへの適用は容易ではない。

### 2.3 粒子ボリュームレンダリング

ボリュームレンダリングの手法としてはレイマーチングが主流であるが、近年では可視化分野において、粒子ベースボリュームレンダリングが開発された<sup>9)11)</sup>。また CUDA による高速化<sup>10)</sup>も提案されている。これらの研究では、特に精度が重要であるため、ノイズを抑えるために、大量の粒子を事前生成しておくことを前提としている。現在のところ、CG を目的とした粒子ボリュームレンダリングは報告されていないが、火炎は高速かつ複雑な反応であり、人間の視覚系がはっきりととらえることは困難であるため、火炎のレンダリングにおいてはノイズがある程度発生することはむしろ好都合であると考えられる。

## 3. アプローチ

上述の通り、現在までのところ、手続き的なインタラクティブ火炎レンダリング手法が提案されていない。これを踏まえ、以下の特長を有する火炎シミュレーション法を提案する。その主たる特長は汎用性、高速性、調整性であり、それぞれ詳細を以下に述べる。

### 3.1 汎用性

プログラマブルシェーダの進歩に伴い、複雑化するレンダリング技法の中で、既存 API の活用や入力データへの要求が少ないことは実用性の点で重要である。提案手法のレンダリングは主に Z バッファ法とアルファテスト、ポイントスプライトといった標準的なレンダリ

ング手続きから構成され、これらは API によるサポートがなされるため実装が容易である。また、提案手法は、入力データとして粒子座標と温度のみを要求し、特定のシミュレーション手法やアニメーション手法に依存しない。

### 3.2 高速性

高速にレンダリングが行えれば、バーチャルリアリティやシミュレータのようなリアルタイムグラフィクスへの適用はもちろん、プリレンダリングムービーの作成においてもアーティストの試行錯誤の労力を減らすことができる。提案手法はソートが不要であり、また、上述の通り標準的なレンダリング手続きから構成されるため、ハードウェアによる高速化がなされている。これらから、提案手法は充分な高速性を保持することができる。

### 3.3 調整性

火炎は想定する現象やモニター、静止画、動画圧縮形式、またそれを知覚する人間で大きく印象が異なる。従ってどのようなレンダリング手法を用いても完全にはなり得ないが、そのため調整の容易性は他の自然表現よりも重要である。提案手法は少数のパラメータで、激しい炎や安定的な炎の表現、速度とレンダリング品質のトレードオフなどが容易に調整できる。

## 4. レンダリングモデル

前節の要請を満たすため、粒子ベースボリュームレンダリングをベースとして用い、それをインタラクティブ火炎に適用させることとした。なお、以下では混乱を避けるため、入力される粒子を物理粒子、物理粒子から乱数によって生成される描画用の一時粒子を描画粒子と定義する。また、Z バッファ法によるレンダリング回数をリピートレベルとする<sup>10)11)</sup>。

### 4.1 粒子ベースボリュームレンダリング

本項では粒子ベースボリュームレンダリングの枠組みを紹介する。詳細は各報告<sup>9)10)11)</sup>を参照のこと。格子データから各セルに格納されているスカラー値に応じて粒子を事前生成する。描画粒子生成座標は乱数によって決定される。この際の描画粒子数と描画粒子座標は隣接セルのスカラー値を参照することで滑らかになる。生成された描画粒子を複数回 Z バッファ法でオフスクリーンレンダリングを行うことによって擬似的にアルファ値を再現する。同報告では最大二百万程度の四面体格子から、数十万から数百万の描画粒子を事前生成し、それをリピートレベル数十から百程度で表示している。

これらの研究は可視化を前提としているため、精度を向上させるために粒子ベースボリュームレンダリングの粗さを抑えなくてはならない。一方本研究ではインタラクティブな火炎表

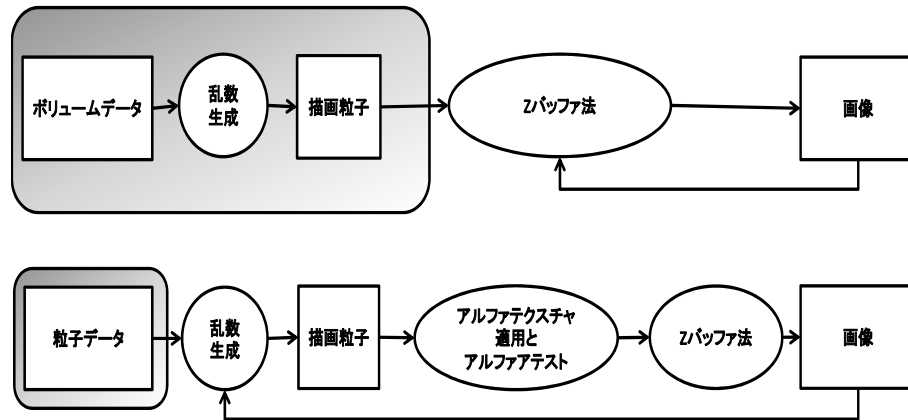


図 1 粒子ベース火炎レンダリングモデル概観。粒子ベースボリュームレンダリングを火炎特性に合わせて合理的に簡略化する。上は元となる粒子ベースボリュームレンダリング、下は提案する粒子ベース火炎レンダリング。枠線部は事前処理を行う箇所である。本手法は粒子入力のみが事前処理である。

Fig.1 Overview of particle-based flame rendering model. Our method is a simplified version of the particle-based volume rendering in consideration of flame characteristics. The original particle-based volume rendering model (top) and our proposed particle-based flame rendering model (bottom). Framed parts depict preprocessing stages. Our model requires only particle input as its preprocessing.

現が目であり、視覚的にそれらしく見ることが可能かどうかという点が重要なため、描画粒子生成を火炎用に簡易化し、アルファテクスチャとアルファテストにより描画粒子数を抑え、色転送関数に黒体放射モデルを用いることでこれを達成する。

既存の粒子ベースボリュームレンダリングと提案手法である粒子ベース火炎レンダリングの違いの概要を図 1 に示す。

#### 4.2 描画粒子生成

粒子ベースボリュームレンダリングでは、隣接格子から描画粒子を生成するため、一様サンプリング法の他、格子内の密度変化を考慮して棄却サンプリング法、メトロポリスサンプリング法などを用いることができる。本研究では接続情報のない物理粒子から描画粒子を生成するため、一様サンプリングを用いることとする。したがって、物理粒子の占めている空間に一様乱数で描画粒子を生成する。この際の乱数値は乱数テーブルを事前に作成しておくことでより高速に処理される。また、各フレームにおいて相対位置を保存しておかないとフレーム間の連続性が途切れてしまうため、最初に乱数テーブルを用意しておき、それを毎フ

レーム参照していくものとする。事実上のアルファ値に相当する生成粒子数は対象が火炎であるため、温度に比例させる。

#### 4.3 アルファテストとアルファテクスチャ

アルファテストとは、一定値未満のアルファ値のピクセルを無いものとして処理する技術である。本研究ではアルファテクスチャを用意し、ポイントスプライトを用いて各描画粒子に貼りつける。アルファテストとアルファテクスチャを用いることで、粒子ベースボリュームレンダリングに必要な、背景が透ける状態を 2.5 次元的に再現する。通常は、ポイントスプライト用のテクスチャはアルファ値を放射状に用意するが、そのままレンダリングを行うと、各描画粒子が球状にはっきりと見えてしまうため、極めて小さな粒子が必要となる。本研究ではこれへの代替策として、放射状のアルファ値に Perlin Noise<sup>7)</sup> を乗算して、穴あきアルファテクスチャを用いた。

#### 4.4 色転送関数

粒子ベースボリュームレンダリングはボリュームレンダリングの汎用的な手法であるため、色転送関数に関して特別な指定がない。本研究で扱う対象は火炎であるため、色転送関数は黒体放射による Nguyen らの手法<sup>6)</sup> と同様の計算手法で定めた。この手法を用いて、3,000K を最大温度とした色伝達関数を図 2 に示す。

#### 4.5 擬似コード

レンダリングプログラムの擬似コードを以下に示す。

```

for all repetitionLevel do
  for all physicsParticles do
    generatePoints
    for all generatedPoints do
      renderPoints
    end for
  end for
  renderToBuffer(1.0/repetitionLevel)
end for
returnFromBuffer(intensityScale)

```

毎回のレンダリング結果はアキュムレーションバッファに格納する。実際には、レンダリング総和をとると、輝度が落ちやすくなるため、最後に輝度補正を行うことが望ましい。



図 2 粒子の温度に対する色を決定する伝達関数。3000K を白色と仮定しており、低温になるにしたがって赤成分が多くなる。

Fig.2 Color transfer function used for associating a color with the temperature of each particle. Color white is associated with the temperature of 3000K. Red component increases as the temperature decreases.

## 5. 調 整

3 節で述べた通り、火炎は視覚的印象が大きく異なる現象であり、また火炎にも様々な種類があるため、要求される状況に応じた最終調整は避けて通れない。本手法ではいくつかの調整パラメータが必要となるが、優先度の高いと考えられる順に詳細を示す。

### 5.1 リピートレベル

リピートレベルが多いほど描画精度は向上する。ただし Z バッファ法はハードウェア負荷が高い処理の一つであるため、要求されるフレームレートや使用している GPU に応じて調整する必要がある。人間自身の残像効果も強く発生するため、リアルタイムグラフィクスへの適用を目的とする場合は比較的少ないリピート回数でも、高速にレンダリングが行われると火炎らしさが生成しやすい。比較画像を 図 3 に示す。

### 5.2 描画粒子のアルファ値

本研究では描画粒子のアルファ値は物理粒子の温度に比例するものとする。比例定数はユーザ指定とする。ここでのアルファ値はアルファテクスチャに乗算する。本手法ではアルファブレンディングを行うわけではなく、アルファテストへの合格率という形に集約されるが、実質不透明度と等価な処理となる。

### 5.3 描画粒子のサイズ

描画粒子は小さい方がディテールを表現しやすいが、描画粒子が小さいと空間を充填するためには大量の描画粒子生成が必要となり、これは GPU への負荷が高い。これも両立を図れるようにポイントのサイズを指定する。

### 5.4 アルファテスト閾値

閾値が低いと、多くのピクセルが合格してしまうため、背後の描画粒子が描画されにくくなり、球状の印象が強くなる。閾値が高いと、多くのピクセルが不合格となるため、ノイズ

のような印象を生成する。ノイズは火の粉のような印象を発生させるため、この調整は、感覚的には、火の粉の多寡の調整に近い。比較画像を 図 4 に示す。

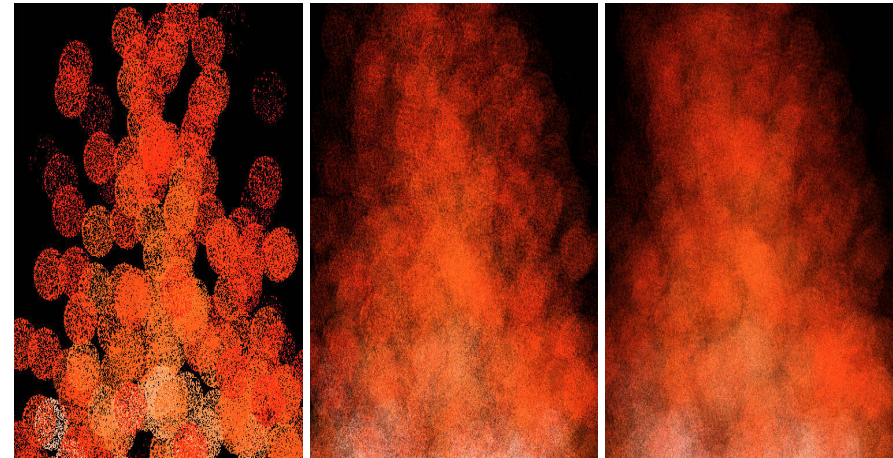


図 3 リピートレベルによる比較。左から繰り返し回数 1, 繰り返し回数 10, 繰り返し回数 20 である。繰り返し回数が増えるに従い、粒子から構成されている印象が薄くなる。

Fig.3 Comparison of rendered flames with different repetition levels 1, 10 and 20 (left to right). Granular effects become unnoticeable as the repetition level increases.



図 4 単一粒子を用いたアルファテスト閾値による比較。左から閾値 0.2, 0.4, 0.6, 0.8 である。閾値が高いほど背後のピクセルが表示されやすくなる。

Fig.4 Comparison of a particle with different alpha test thresholds 0.2, 0.4, 0.6 and 0.8 (left to right). Background pixels appear more frequently as the threshold increases.

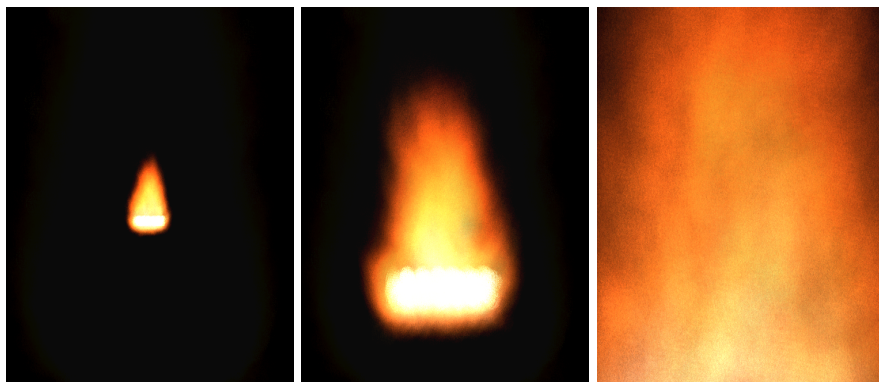


図 5 視点距離による比較. 視点距離が短いほど, デティールが付加される. この際に, パラメータ調整は不要である.  
Fig. 5 Comparison of rendered flames with different viewing distances. The detail gets added as the viewing distance becomes smaller. Note that the parameter adjustment is unnecessary.

## 6. 結 果

本稿中の結果は CPU に Intel Core2 2.40GHz, GPU に nVidia GeForce 8800, 実装に C++, グラフィクス API に OpenGL を用いて実装したものである. なお, OpenGL の仕様は 1.5 を用い, プログラマブルシェーダは用いなかった.

視点距離に応じたレンダリング結果を比較した画像を 図 5 に示す. これらは同一パラメータであるにもかかわらず, 視点距離が長い場合には遠くで燃えている印象が, 視点距離が短い場合には燃え盛っている印象が生じている. これは通常のポイントスプライトと違い, Z バッファ法を用いており, ピクセル単位の処理が支配的となっているためである.

火炎の粒子生成として, 乱数をベースとしたものを 図 6 に示す. これは事前計算となる粒子生成コストが少ないが, 動きが単調になりやすい. また, 粒子生成として, SPH<sup>13)</sup> を用い, 空と地面をテクスチャとして用意し, これに本手法による火炎レンダリングを行った画像を 図 7 に示す. これは粒子生成コストがやや大きくなるが, 動きに現実性がある. この際の物理粒子数は 2,500, 描画粒子数は 50,000 程度であり, レンダリングは約 0.03 秒で完了した. 以上より, 本手法は十分なインタラクティブ性を保持しているものとする.

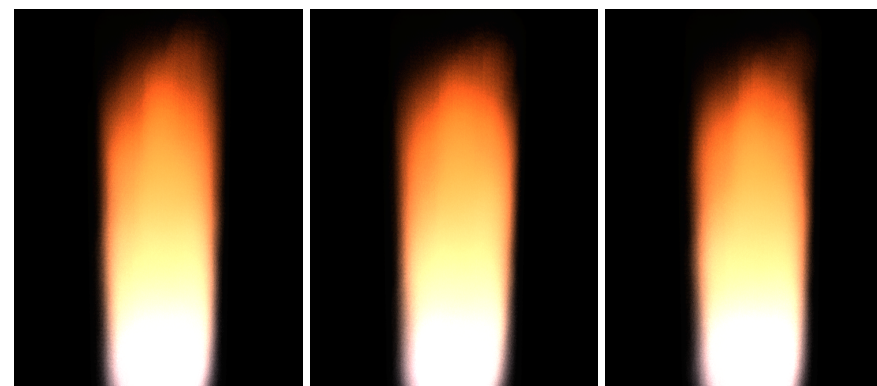


図 6 乱数ベースによるレンダリング結果. 滑らかではあるが, 動きが単調となりやすい.  
Fig. 6 Rendering of random number-based particle simulation. A smooth, but monotonous flame is generated.

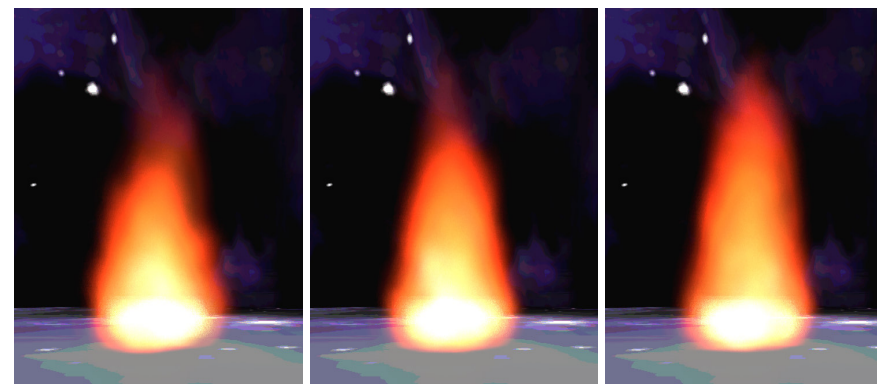


図 7 粒子シミュレータに SPH を適用し, さらに背景にテクスチャを用いたレンダリング結果. 乱数ベースと異なり, 揺らぎが発生する.  
Fig. 7 Rendering of SPH-based particle simulation with a background texture. Flickers can be generated unlike the result in Figure 6.

## 7. 結論と今後の課題

本稿では粒子ベースボリュームレンダリングを応用したインタラクティブ火炎レンダリング技法を提案した。結果として、従来の描画法では難しい火炎のディテールを、ソート処理を行うことなく、既存 API を活用し、少数の粒子から作成することができた。

プログラマブルシェーダを用いた実装により表現の幅を広げることが可能であると考えられるが、自由度が向上する反面、調整係数が増加してしまうためかえって扱いにくくなるとも考えられ、このトレードオフをどう扱うかの調整は UI を含めて必要なものとなると考えられる。

今後の表現対象として、小規模火炎だけでなく、煙、霧のような対象や、大規模爆発火炎への応用が挙げられる。

## 参 考 文 献

- 1) Carpenter, L.: The A-buffer, an Antialiased Hidden Surface Method, *Computer Graphics*, Vol. 18, No. 3, pp. 103–108 (1984).
- 2) Horvath, C. and W, G.: Directable, High-Resolution Simulation of Fire on the GPU, *ACM Transactions on Graphics*, Vol. 28, No. 3, Article 41 (2009).
- 3) Inakage, M.: A Simple Model of Flames, *Proceedings of CG International '90*, pp. 71–81 (1990).
- 4) Lamorlette, A. and Foster, N.: Structural Modeling of Flames for a Production Environment, *ACM Transactions on Graphics*, Vol. 21, No. 3, pp. 729–735 (2002).
- 5) Myers, K. and Bavoil, L.: Stencil Routed A-Buffer, *SIGGRAPH 2007 sketches* Article 21 (2007).
- 6) Nguyen, D., Gibou, F. and Fedkiw, R.: Physically Based Modeling and Animation of Fire, *ACM Transactions on Graphics*, Vol. 21, No. 3, pp. 721–728 (2002).
- 7) Perlin, K.: Improving Noise, *ACM Transactions on Graphics*, Vol. 21, No. 3, pp. 681–682 (2002).
- 8) Reeves, W. T.: Particle Systems – A Technique for Modeling a Class of Fuzzy Objects, *ACM Transactions on Graphics*, Vol. 2, No. 2, pp. 91–108 (1982).
- 9) Sakamoto, N., Nonaka, J., Koyamada, K. and Tanaka, S.: Particle-Based Volume Rendering, *Proceedings of Asia-Pacific Symposium on Visualization*, pp. 141–144 (2007).
- 10) Zhongming, D., Kawamura, T., Sakamoto, N. and Koyamada, K.: Particle-Based Multiple Irregular Volume Rendering on CUDA, *Simulation Modelling Practice and Theory*, Vol. 18, No. 8, pp. 1172–1183 (2010).

- 11) 小山田 耕二, 坂本 尚久: 粒子ボリュームレンダリング, コロナ社 (2010).
- 12) 井村 誠孝, 稲垣 智, 池田 聖, 眞鍋 佳嗣, 大城 理, 千原 國宏: 「粒子ベース流体シミュレーションに基づく炎のリアルタイムレンダリング」, 第 15 回日本バーチャルリアリティ学会大会論文集, pp. 560–563 (2010).
- 13) 間淵 聡, 藤代 一成, 大野 義夫: 「SPH ベース火炎シミュレーション」, 情報処理学会研究報告, Vol. 2010-CG-140, No. 9 (2010).