

# 任意形状の爆発アニメーション生成システム

森田 拓也<sup>1,a)</sup> 土橋 宜典<sup>1,2,b)</sup> 山本 強<sup>1,c)</sup>

**概要:** 近年, 流体シミュレーションを利用した映像生成が盛んに行われている. 本稿ではとくに爆発現象の映像生成に着目する. 流体シミュレーションを利用した爆発現象の映像生成を行う手法はいくつか開発されているが, いずれにおいても多くのシミュレーションパラメータを設定する必要があり, 所望する形状を持つ爆発の生成は困難である. そこで本研究では, 所望の爆発アニメーションを容易に生成することができる制御システムを提案する. 提案手法では, 爆発規模最適化ステップと爆発形状編集ステップの2段階による制御を行う. 提案システムを適用することで, ユーザが所望する形状に従って制御された爆発アニメーションを生成することができる.

**キーワード:** 流体シミュレーション, 爆発現象, 最適化

## A System for Generating Animation of Arbitrary Shape Explosion

TAKUYA MORITA<sup>1,a)</sup> YOSHINORI DOBASHI<sup>1,2,b)</sup> TSUYOSHI YAMAMOTO<sup>1,c)</sup>

**Abstract:** Recently, many methods for creating fluid animation based on fluid simulation have been proposed. We focus on the simulation of explosions. Some methods for simulating explosion were already developed. Although these methods can create realistic shapes and movement of explosions, the resulting shapes and motion depend on many simulation parameters. Therefore, it is very difficult to create the explosions with desired shapes. In this paper, we propose a system for controlling explosion shapes towards desired shapes. Our system consists of the following two steps: scale-optimization step and shape-editing step. Using the proposed system, the animation of explosions controlled into specified shapes can be generated.

**Keywords:** Fluid simulation, Explosion phenomena, Optimization

### 1. はじめに

コンピュータグラフィックス (CG) がもつ表現力は著しく成長を遂げており, あらゆる物体・現象を対象とした描写においてその写実性が向上している. これらの技術進歩は, 映画やゲームをはじめとするビジュアルコンテンツにおいて確認することができる. 近年においては流体の CG 表現に関しても, 煙や水, 炎などを筆頭に様々な手法が研究されている [1]. とくに流体力学に則った物理シミュレ-

ーションがベースとなる手法では, 非常に写実的な挙動をもつ映像が生成でき, 昨今の CG 制作アプリケーションにおいてもこれらの技術が活用されている. CG による流体映像は屋内外を問わず様々なシーンにおいて活躍し, その高い写実性や動きの複雑さなどから映像に大きなインパクトを与えている.

爆発現象も流体の一つに分類される. 映像作品において爆発現象を取り扱う場合, 実写による再現では複雑な環境構築や撮影の危険性を伴うため, CG が利用されるケースも多い. CG を利用した爆発映像の生成においては, 従来研究として流体解析を用いた手法が提案されている [2], [3], [4]. これらの研究により, 非常に写実的な挙動の爆発アニメーション生成が実現された. しかし, シミュレーションベースの映像生成手法では, 煩雑なパラメータ設定を要すると

<sup>1</sup> 北海道大学  
Hokkaido University, Sapporo, Hokkaido 060-0814, Japan

<sup>2</sup> 独立行政法人科学技術振興機構 CREST  
JST CREST

a) morita@ime.ist.hokudai.ac.jp

b) doba@ime.ist.hokudai.ac.jp

c) yamamoto@ist.hokudai.ac.jp

いう問題点がある。シミュレーションのパラメータ調整を行うためには爆発シミュレータの仕組みを熟知する必要があり、これはユーザにとって大きな負担となる。さらに、パラメータ調整のみでは複雑な形状の爆発映像を生成することは困難である。映像作品においては、シーンを脚色する目的でしばしば非現実的な挙動の爆発表現が用いられるが、シミュレーションのみではこれに対応することができない。

そこで本研究では、シミュレーションベースで任意形状の爆発アニメーションが生成可能なシステムを提案する。本システムでは、シミュレーションパラメータの自動調整による爆発規模の最適化(爆発規模最適化ステップ)、およびポテンシャル制御による爆発形状の編集(爆発形状編集ステップ)によって、所望形状の爆発アニメーションを生成する。爆発規模最適化ステップでは、指定したタイムステップにおけるシミュレーションの密度分布を目標規模に到達させるために、浮力係数および爆発源のポテンシャル強度に対して最急降下法を適用することで最適なパラメータを求める。爆発形状編集ステップでは、シミュレーションにおける全粒子の経路を記憶し、指定した粒子に対応する経路をトレースするようにポテンシャルを付加することで、粒子運動の制御を行う。

## 2. 関連研究

本節では、本研究と深く関連している爆発のビジュアルシミュレーション手法、および流体シミュレーションの制御手法について紹介する。

### 2.1 爆発のビジュアルシミュレーション

CG分野において初めてシミュレーションベースによる爆発映像の生成手法を提案したのはYngveらである[2]。Yngveらの手法では圧縮性の流体方程式を解くことで、爆発現象とその周辺の衝撃波の映像を再現することができる。しかしこの手法では非常に高い計算コストが要求されてしまう。これに対しFeldmanらが提案した手法では、より簡易的な計算で爆発映像を生成することによって、計算コスト面の問題を改善している[3]。この手法では、流体解析としてStamらが提案した格子法によるNavier-Stokes方程式(N-S方程式)の安定解法[5]を使用し、燃焼作用を表す粒子との相互作用を考慮することで、爆発や液体スプレーの燃焼などを映像化している。Rasmussenらの手法では、大規模の爆発を高精細かつ低コストで再現するために、2次元の爆発シミュレーションを組み合わせることで映像の3次元化を図っている[4]。いずれの手法においてもシミュレーション中で煩雑なパラメータ調整が必要となり、ユーザが意図する規模・形状をもつ爆発映像を作成するのは極めて困難である。

### 2.2 流体シミュレーションの制御

流体シミュレーションに関しては多くの制御手法が提案されており、パラメータ調整における作業負担の軽減や、シミュレーションのみでは再現が困難なアニメーションの生成に挑戦している。Treuilleらが提案した手法では煙のシミュレーションを対象としており、指定したタイムステップにおいて指定した密度分布に近づくような外力場を形成する[6]。これに対してFattalらの手法では、N-S方程式の外力項に対して、目的形状に近づけるためのdriving force termと、密度場の拡散を抑えるためのgathering termを付加することで形状制御を行う[7]。Dobashiらは積乱雲の発達後形状に着目し、その発達を促す潜熱係数のフィードバック制御を行うことで所望のアニメーションを生成する[8]。これまでに挙げた制御手法では爆発現象は考慮されていない。文献[6]、[7]のような外力による制御を行うと、シミュレーションにより得られた爆発らしい挙動が失われてしまうため、文献[8]のように対象とする流体の特性に基づいた制御を行うことが望ましい。

爆発シミュレーションの制御手法に関しては佐藤らの手法が挙げられる[9]。この手法では、複数の2次元爆発シミュレーションに対し爆発源のパラメータを調整し、目標形状への到達度を評価することで制御を行う。さらに、文献[4]の手法により複数の2次元密度場から3次元密度場を合成することで、所望の3次元形状をもつ爆発アニメーションを生成することができる。しかし、この手法では流体の圧縮性は考慮されておらず、外力付加による疑似的な爆発シミュレーションを対象としている。

本手法では流体の圧縮性を簡易的に再現した3次元爆発シミュレーションに対して制御を行うことで、所望形状をもつアニメーションの生成を目的とする。これにより、シミュレーションの自然な挙動を維持したまま任意形状を形成することを目指す。

## 3. シミュレーション手法

本稿における爆発シミュレータは、文献[3]を参考に簡易的な解析モデルを構築している。以下でシミュレーション手法について詳しく説明する。本手法では、まず3次元空間中で爆発が起こった際の状態に関して、周りの空気も含めた解析を行う。この解析には格子法を用いており、空間中を格子数 $N_x \times N_y \times N_z$ 、格子間隔 $dh$ で分割し、各格子点に対して速度、温度、密度の解析情報を割り当てる。シミュレーション空間中には浮遊する粒子が多数配置されており、解析した速度場に応じてそれぞれが流動する。これらの浮遊粒子を描画することによって爆発アニメーションが生成される。

また、爆発を発生させるためにシミュレーションの初期状態として空間中で指定した領域に爆発源が配置されている。爆発源では密度 $D_{src}$ 、温度 $T_{src}$ 、およびポテンシ

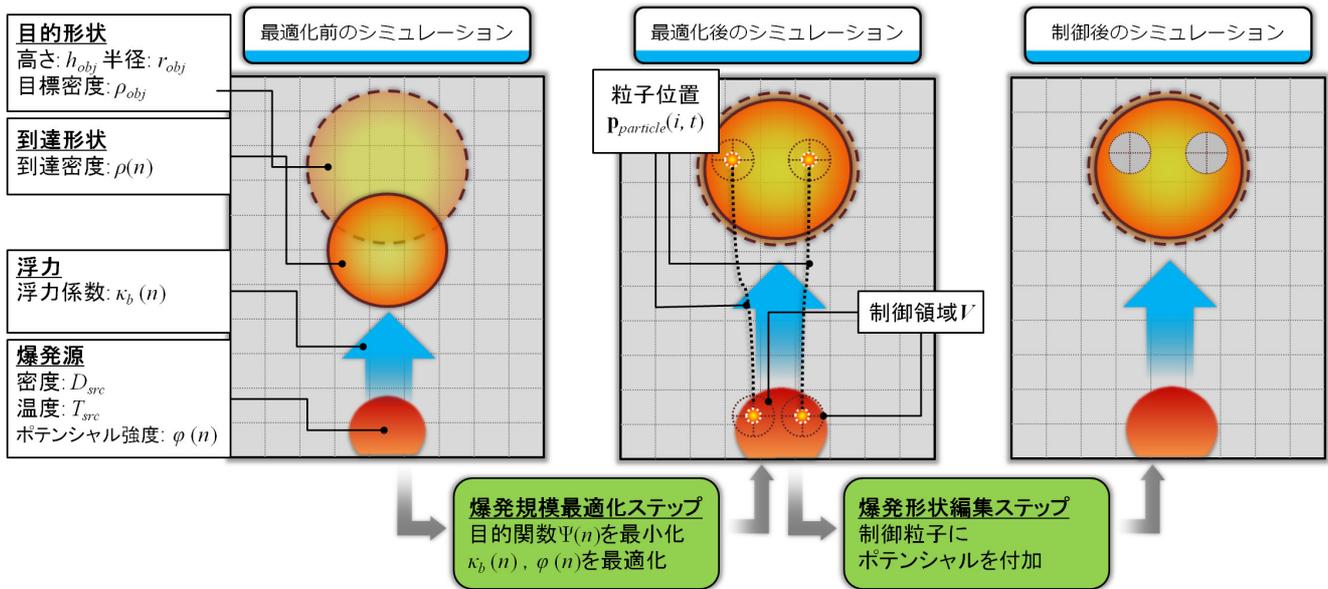


図 1 提案システムの概要

ル強度  $\phi$  が割り当てられており、シミュレーションの各タイムステップにおいてそれぞれの値が指定した領域に加わることによって爆発現象が発生する。なお、爆発現象は瞬間的に発生する性質があるため、爆発源の値は時間経過に伴って減衰する。

流体解析処理における支配方程式は次に示す N-S 方程式に従う。本稿では簡略化のため非粘性流体を仮定する。

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

ここで、 $\mathbf{u}$  は流体の速度ベクトル、 $\rho$  は流体の密度、 $p$  は圧力である。 $\mathbf{f}$  は外力全般を表しており、重力や浮力などの様々な影響を考慮している。この式は流体の時間発展を表す方程式であり、右辺第 1 項から移流項、圧力項、外力項を指す。爆発現象に関しては圧縮性流体の特性を考慮する必要があるため、速度場の発散に関して次式に示す条件を設ける。

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

$\phi$  は爆発源においては何らかの値をもち、それを除く領域ではゼロとなる。この条件を簡易的に満たすために、圧力項の解を決定するポアソン方程式に関して次のような修正を加える。

$$\nabla^2 p = \frac{\rho}{\Delta t} (\nabla \cdot \mathbf{u} - \phi) \quad (3)$$

温度場  $T$  の解析については、式 (1) の支配方程式に加え冷却を考慮した次式を用いる。

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

ここで  $c_r$  は冷却定数、 $T_{amb}$  は環境温度である。 $T_{max}$  はシミュレーション空間内における最大温度であり、この値

にはシミュレーションにより得られた温度の最大値を割り当てる。また、温度場から発生した浮力を考慮して、速度場を更新する必要がある。浮力計算は次式の通りである。

$$\mathbf{f}_{buo} = \kappa_b (T - T_{amb}) \mathbf{z} \quad (5)$$

ここで  $\kappa_b$  は浮力係数、 $\mathbf{z}$  は鉛直方向の単位ベクトルである。

以上の方程式の数値解析を高速化するため、本手法では Crane らが考案した GPU 処理モデル [10] を用いる。さらに、乱流成分を疑似的に付加するため Fedkiw らの手法 [11] をもとに次式の渦補正力も外力項に加える。

$$\mathbf{f}_{conf} = \epsilon (\mathbf{N} \times \boldsymbol{\omega}) \quad (6)$$

ここで、 $\epsilon$  は渦補正係数を表し、 $\mathbf{N}$  は  $\boldsymbol{\omega} = \nabla \times \mathbf{u}$  を用いて表現すると  $\mathbf{N} = \nabla |\boldsymbol{\omega}| / |\nabla |\boldsymbol{\omega}||$  となる。

#### 4. 制御手法

提案システムの概要を図 1 に示す。本システムは、大きく爆発規模最適化ステップと爆発形状編集ステップの 2 段階に分かれている。

爆発規模最適化ステップでは、まずユーザが任意のタイムステップ  $t_{obj}$  と所望する爆発規模を指定する。爆発規模の指定には高さ  $h_{obj}$  および半径  $r_{obj}$  を用い、 $\mathbf{p}_{obj} = dh(\frac{N_x}{2}, \frac{N_y}{2}, h_{obj})$  を中心として半径  $r_{obj}$  の球状の密度分布  $\rho_{obj}$  を目標形状とする。タイムステップ  $t_{obj}$  において、シミュレーションによって計算される密度分布  $\rho$  が  $\rho_{obj}$  と一致するよう、パラメータの最適化を行う。最適化を行うパラメータは、爆発の到達地点に影響を与える浮力係数  $\kappa_b$ 、および爆発の大きさに影響を与えるポテンシャル強度  $\phi$  である。最適化は最急降下法により行い、以降は最適化試行回数  $n$  における浮力を  $\kappa_b(n)$ 、ポテンシャル強度

を  $\phi(n)$ , 密度分布を  $\rho(n)$  と表記する.

次に, 爆発形状編集ステップへ移行する. まず, 爆発規模最適化ステップで決定されたパラメータを用い, 爆発のシミュレーションを行うとともに, 爆発源付近に多数の粒子を生成し, その移動軌跡を記憶しておく. いま, タイムステップ  $t$  における粒子  $i$  の位置を  $\mathbf{p}_{particle}(i, t)$  と表す. ユーザは, タイムステップ  $t_{obj}$  における粒子分布に対して, 編集操作を施す. 具体的には, ユーザが指定した領域の内部に粒子が侵入しないよう指定する. これをもとに, 再びシミュレーションを行い, 以下のような方法で目的形状となるよう制御を行う. まず, ユーザが指定した領域中心に最も近い粒子を制御粒子として選択する. そして, 制御粒子の周辺に反発力を発生し, 粒子の侵入を防ぐ. 反発力の大きさについては, 周辺粒子との距離をフィードバックすることで決定する.

最後に, 生成された映像をより所望形状に近づけるための補正処理を行う. 以降各ステップについて詳しく説明する.

#### 4.1 爆発規模最適化ステップ

本手法では, 非線形計画法の中でも最急降下法による最適化を行っているため, 目的関数の定義, および対応する各パラメータで目的関数を微分した値が必要となる. そこで,  $n$  回の最適化を行ったシミュレーションの密度分布  $\rho(n)$  と目標形状の密度分布  $\rho_{obj}$  との類似度を示す目的関数  $\Psi(n)$  を次式の通り定義する.

$$\Psi(n) = k_{\psi}(\rho(n) - \rho_{obj})^2 \quad (7)$$

$$\rho_{obj}(i, j, k) = \begin{cases} D_{src} & (|\mathbf{p}(i, j, k) - \mathbf{p}_{obj}| \leq r_{obj}) \\ 0 & (otherwise) \end{cases} \quad (8)$$

ここで,  $k_{\psi}$  は調整用係数で, ユーザにより指定する定数である.  $(\rho(n) - \rho_{obj})^2$  は, それぞれの密度場に対応する格子点同士の差分を取り, その二乗誤差の総和値を計算する.  $\mathbf{p}(i, j, k)$  は, 格子点  $(i, j, k)$  の座標を表す. 以上のように目的関数を構築することで, シミュレーションの密度分布が目標形状に近いほど関数値は小さくなるので, この値を最小化するように浮力とポテンシャル強度の最適化を行う. 試行回数  $n+1$  における各パラメータの更新値は次式に従う.

$$\kappa_b(n+1) = \kappa_b(n) - \alpha \frac{\partial \Psi(n)}{\partial \kappa_b(n)} \quad (9)$$

$$\phi(n+1) = \phi_k - \beta \frac{\partial \Psi(n)}{\partial \phi(n)} \quad (10)$$

$$\frac{\partial \Psi(n)}{\partial \kappa_b(n)} = \frac{\Psi(n) - \Psi(n, \Delta \kappa_b)}{\Delta \kappa_b} \quad (11)$$

$$\frac{\partial \Psi(n)}{\partial \phi(n)} = \frac{\Psi(n) - \Psi(n, \Delta \phi)}{\Delta \phi} \quad (12)$$

$\alpha$  および  $\beta$  は最急降下法における各パラメータを更新する際の刻み幅である.  $\Psi(n, \Delta \kappa_b)$  および  $\Psi(n, \Delta \phi)$  は,  $\Psi(n)$  に対して, それぞれのパラメータ値を微小量変化させたときの目的関数の値である.

#### 4.2 爆発形状編集ステップ

爆発形状編集ステップでは, シミュレーション空間中に爆発源とは異なるポテンシャル場  $\phi^*$  を重畳することで粒子の挙動を制御する.  $\phi^*$  は各タイムステップ毎に更新され, 制御粒子の経路に沿ってポテンシャルが付加される.  $i$  番目の制御粒子の座標において, タイムステップ  $t$  の時点で付加されるポテンシャル強度  $\phi^*(\mathbf{p}_{particle}(i, t))$  は次式に従う.

$$\phi^*(\mathbf{p}_{particle}(i, t)) = w_c^3 \frac{n_{particle}}{V} \quad (13)$$

$$w_c = \begin{cases} \frac{t}{t_{obj}} & (t < t_{obj}) \\ 1 & (otherwise) \end{cases} \quad (14)$$

ここで,  $w_c$  はポテンシャルを適用する度合いを表しており, 目標タイムステップに近づくに連れて値が 1 に近づく. この係数により時間経過に伴うポテンシャル強度の増加が実現できる. また,  $V$  は指定した制御領域の体積で,  $n_{particle}$  は制御領域内に存在する粒子数である. これにより, ユーザが指定した制御領域内に存在する粒子量に応じたフィードバック制御が可能となる.

なお, このステップでは複数の制御粒子を指定することが可能であり, これらを組み合わせることで複雑な形状の爆発アニメーションを生成することができる.

#### 4.3 爆発形状の補正

爆発形状編集ステップで指定した制御形状によっては, 編集処理後の形状が所望する形状に収束しない問題が発生する. 本システムではこれを解決するために, 編集処理後における粒子の挙動に対して次式に示す補正処理を加える.

$$\mathbf{p}_{particle}^*(i, t) = \mathbf{p}_{particle}^*(i, t) + w_r(\mathbf{p}_{particle}^*(i, t) - \mathbf{p}_{particle}(i, t)) \quad (15)$$

ここでは, 編集処理前の粒子の位置を表す  $\mathbf{p}_{particle}(i, t)$  に対して, 編集処理後の粒子の位置を表すベクトルとして  $\mathbf{p}_{particle}^*(i, t)$  を定義する. そして, 対応する粒子番号およびタイムステップにおけるそれぞれの座標間の距離に応じて,  $w_r$  による重みづけを加えた上で補正処理を施している.

この処理は流体解析後の浮遊粒子に対して後処理として適用するため, シミュレータそのものには影響を与えない.

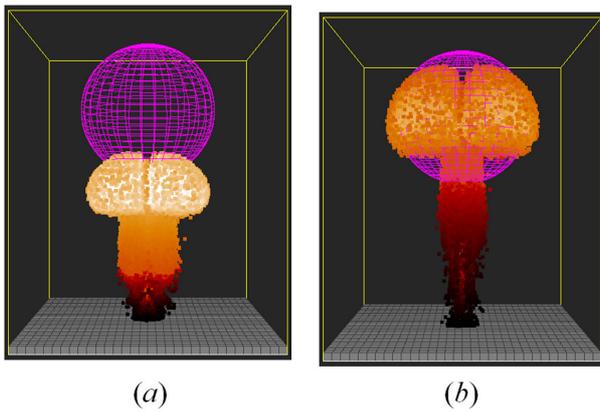


図 2 爆発規模最適化ステップの実験結果

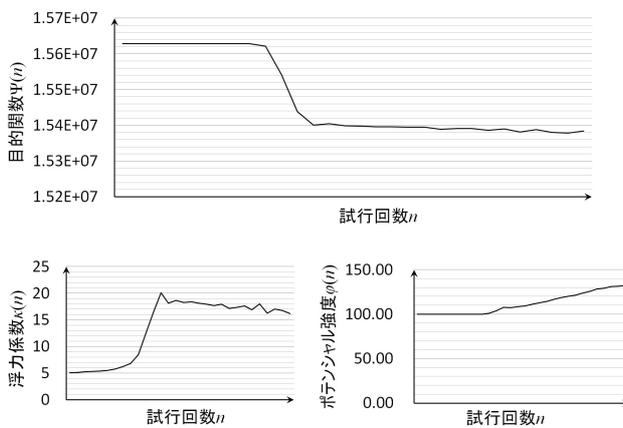


図 3 目的関数と対応するパラメータの推移

## 5. 実験

本節では、提案システムを用いて爆発アニメーションを生成する。以下で本稿で行った実験について詳しく説明する。まず、初期パラメータを設定した爆発シミュレーションを行う。ここで、シミュレーションに用いる格子数は  $32 \times 32 \times 40$ 、格子幅  $dh$  は 1.0、1 ステップ毎の時間間隔は  $dt$  は 0.01 である。また、爆発源は座標 (16, 16, 3) 地点を中心とした半径 3.0 の範囲内に温度  $T_{src} = 150.0$ 、密度  $D_{src} = 5.0$ 、浮力係数  $\kappa_b(0) = 5.0$ 、初期ポテンシャル強度  $\phi(0) = 100.0$  を設定する。空間中に配置する粒子数は 40,000 で、爆発源の範囲内にランダムで配置する。以降の実験では、タイムステップ  $t_{obj} = 120$  を対象とした制御を行う。

まず、爆発規模最適化ステップの実験結果について論じる。この実験における目標形状は高さ  $h_{obj} = 30.0$  に中心がある半径  $r_{obj} = 8.0$  の球とする。図 2 に実験結果を示す。最適化を高速に行うために、このステップの結果描画においては色つき粒子を直接描画するのみである。目標形状はシミュレーション空間中に紫色で示している。(a) は初期パラメータでシミュレーションを行った結果である。この状態では、爆発の大きさや高さが目的形状に到達していな

いことがわかる。これに対し、提案システムによる爆発規模の最適化を行った結果が (b) である。(b) では目標形状内に収まるような爆発形状となっていることがわかる。図 3 に最適化試行回数毎の目的関数値、およびパラメータの推移を示す。本稿における最適化試行回数は 30 回であり、最終的なパラメータ値は  $\kappa_b(30) = 16.21$ 、 $\phi(30) = 132.34$  となった。なお、シミュレーション 1 ステップの処理時間は平均 32[m $s$ ]、最適化 1 回の処理時間は平均 20.76[s] となった。

次に、爆発形状編集ステップの実験結果について論じる。図 4 に実験結果を示す。このステップにおける結果描画では、スプラッティング法による粒子のボリュームレンダリングを行っている。(a) は編集前の爆発形状である。(b) は編集中の画面で、紫色の円で示した部分が制御領域となる。本実験においては、選択した粒子を中心として  $xz$  平面を垂直に貫く円柱を制御領域としており、この領域の体積は  $V = N_y \pi r_i^2$  となる。ここで、 $r_i$  は制御対象の粒子番号  $i$  毎にユーザーが指定した半径である。(c) は補正処理を行わない場合の実験結果である。この結果では制御領域外の粒子も大きく変化してしまっているが、補正処理を加えた結果である (d) では、制御領域付近の粒子のみが退けられていることがわかる。なお、粒子経路の記憶を 1 ステップ分行うための処理時間は平均 3[m $s$ ]、制御処理を加えたシミュレーション 1 ステップの処理時間は平均 40[m $s$ ] となった。

## 6. まとめと今後の課題

本稿では、爆発シミュレーションに対して爆発規模最適化ステップによるパラメータ調整、および爆発形状編集ステップによるポテンシャル場の形成を行うことで、任意形状の爆発アニメーションを生成するシステムを提案した。また、実験を通じて目標形状を得るための最適な浮力、およびポテンシャル強度の導出が可能であることや、指定した領域に粒子が侵入しないようなアニメーションの生成が可能であることを確認した。

今後の課題としては、シミュレータに対して粒子の燃焼現象を考慮することや、より複雑な形状編集を適用可能にすることで、3次元形状の編集を実現することなどが挙げられる。

**謝辞** この研究は独立行政法人科学技術振興機構、CREST によりサポートされています。

## 参考文献

- [1] R. Bridson, Fluid Simulation for Computer Graphics, AK Peters, September 2008.
- [2] G.D.Yngve, J.F. O'Brien, J.K.Hodgins, "Animating explosions", In Proceedings of ACM SIGGRAPH 2000, 29-36, July 2000.
- [3] B.E.Feldman, J.F. O'Brien, O.Arkan, "Animating Sus-

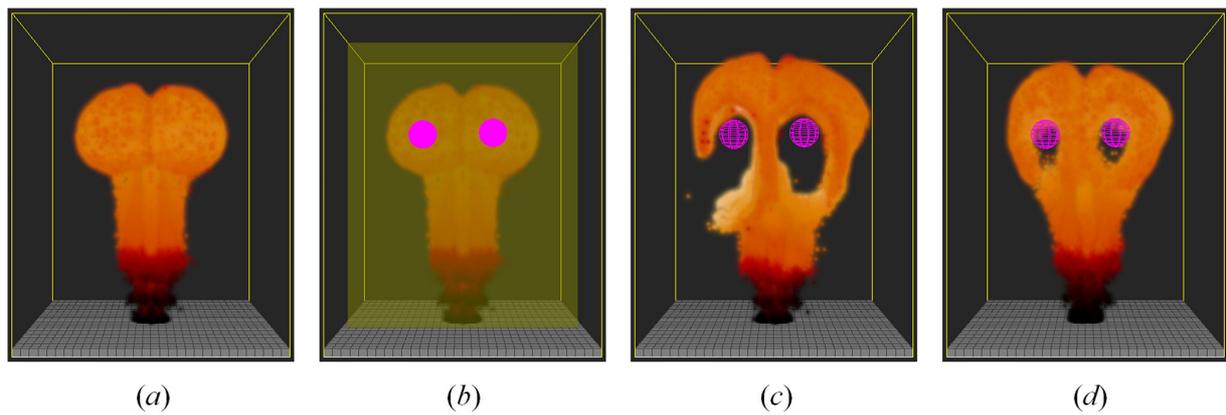


図 4 爆発形状編集ステップの実験結果

- pended Particle Explosions”, In Proceedings of ACM SIGGRAPH 2003, 708-715, Aug 2003.
- [4] N.Rasmussen, D.Q.Nguyen, W.geiger, R.Fedkiw, “Smoke simulation for large scale phenomena”, In Proceedings of ACM SIGGRAPH 2003, 703-703, Aug 2003.
  - [5] J. Stam, “Stable fluids,” In Proceedings of ACM SIGGRAPH 1999, 121-128, Los Angeles, USA, August 1999.
  - [6] A.Treuille, A.McNamara, Z.Popovic, J.Stam, “Keyframe Control of Smoke Simulations”, In Proceedings of ACM SIGGRAPH 2003, 716-723, Aug 2003.
  - [7] R.Fattal, D.Lischinski, “Target-driven smoke animation”, In Proceedings of ACM SIGGRAPH 2004, 441-448, Aug 2004.
  - [8] Y. Dobashi, K. Kusumoto, T. Nishita, T. Yamamoto, Feedback control of cumuliform cloud formation based on computational fluid dynamics, ACM Transaction on Graphics, Article 94, March 2008.
  - [9] 佐藤 周平, 土橋 宜典, 山本 強, 安生 健一, 最適な初期強度分布の推定および予測制御による爆発シミュレーションの制御, 映像メディア学会誌 Vol.65, No.10, pp.1-6, 2011.
  - [10] K. Crane, I. Llamas, and S. Tariq, Real Time Simulation and Rendering of 3D Fluids, Addison-Wesley, ch.30, 2007.
  - [11] R. Fedkiw, J. Stam, and H. W. Jensen, “Visual simulation of smoke” In Proceedings of ACM SIGGRAPH 2001, 15-22, August 2001.