

## アンサンブル平均による非構造格子向けボリュームレンダリング

## A Technique of Irregular Volume Rendering Using Ensemble Average

小山田 耕二

Koji KOYAMADA

## 概 要

本稿ではアンサンブル平均を用いた非構造格子向けボリュームレンダリングについて説明する。大規模非構造格子データ向けのボリュームレンダリングにおいては、視点に沿った方向での格子のソート処理がボトルネックとなる問題が認められている。この問題を解決するためにこれまで提案されたソート不要を特長とする手法では、吸収や自己発光というボリュームレンダリングにとって本質的な効果を見逃していた。我々は、ボリュームデータを表現する発光粒子を画像平面に投影することを繰り返すアンサンブル平均を取り入れることにより、吸収・自己発光効果を実現しながらもソート処理が不要のボリュームレンダリングアルゴリズムを実現した。

## 1. はじめに

雲の表現を使ったボリュームデータの可視化は一般にボリュームレンダリングと呼ばれる。我々の知覚システムにおいて感覚の入力が物理的にもっともらしい現象から生じ、そのうえで入力解釈を行うと仮定すると、ボリュームデータは物理的な現象をもっともらしく表現するメディアによって表現されることが望ましい。たとえば、視覚的類似度が高いので、有限要素法によって計算されたクリーンルーム内の空気汚染度はボリュームレンダリングによって効果的に可視化することができる。一般的にボリュームレンダリングは、ボリュームデータの全体的な特徴をうまく表現することが可能である。ボリュームレンダリングは与えられたボリュームデータを半透明な雲として表現され、その見栄えは、スカラデータから色データへのマッピングを指定する伝達関数によって制御される。この指定は、ユーザが興味のあるデータについては強調のため不透明度を上げ、そうでないところはこれを下げることによって実現される。

米国科学財団 によるViSCレポート<sup>1)</sup>が1987年に発行されてから多くのボリュームレンダリング手法が

提案された。それから20年も経た今でも特に非構造格子向けのボリュームレンダリング手法については可視化研究コミュニティから重要な研究分野であると認識されている。非構造格子は、かならずしも正方格子に限らない小立体によって3次元空間を構成したもので、小立体として4面体や6面体がよく利用される。この形式の格子は計算固体・流体力学で利用されることの多い有限要素法の計算結果によく見受けられる。非構造格子向けボリュームレンダリングは、大きく画素ベースか、または格子ベースの手法に分類される。どちらの手法においても基礎格子として4面体が用いられることが多い。

非構造格子向けボリュームレンダリング手法について多くの効率のよい手法が提案されたが、視線に沿った格子のソート処理が今後も開発を継続する必要がある重要な技術として位置づけられている。Sabella<sup>2)</sup>による粒子密度モデルを採用する限りこのソート処理は避けて通ることができない。しかしながら、この粒子密度モデルが導出された粒子モデルに注目することにより、非構造格子向けの効率のよい手法を設計することが可能である。粒子密度モデルは連続的な性質を持つ一方で、粒子モデルは、離散的な性質を持つ。

本解説では、この粒子モデルに回帰することによ

り開発された格子のソート処理が不要となる非構造格子向けボリュームレンダリング法を説明する。本解説は、以下のように構成される。まず、非構造格子向けボリュームレンダリングに関する関連研究について述べる。次に、粒子モデルから導出された密度発光モデルについて説明し、その次にポアソン分布とボリュームレンダリングで重要な役割を果たす不透明度との関係について述べ、アンサンブル平均を用いた画像生成手法について説明する。提案手法の有効性を説明するために本手法を7100万程度の6面体1次要素、そして2400万程度の4面体2次要素からなる巨大非構造格子ボリュームデータに適用する。

## 2. 関連研究

Roettger ら<sup>3)</sup>は、四面体格子ソートに必要とされるメモリバンド幅が処理速度の上限を決定することを示し、ソート不要のボリュームレンダリング手法の提案を行った。彼らのレンダリングモデルは粒子発光モデルから吸収効果は無視したものである。その適用は炎、霧のような現象の表現に限定される。Csébfalvi<sup>4)5)</sup>もソート不要のボリュームレンダリング法を提案したが、先ほど述べた手法とは逆で、吸収効果のみを考慮しており、いわゆるX線撮影像と同様の画像しか作成できない。

Callahan ら<sup>6)</sup>は、四面体の格子面重心を用いて大雑把なソートを行い、その後、有限の深さを持つフレームバッファを用いて格子面から展開される画素レベルでソートの精度を向上させる統合的ソート手法(HAVS)を開発した。HAVSは1.4M個の四面体格子に対して1.3fpsの計算速度を達成したが、フレームバッファの深さが十分でない場合には描画順序の誤りに基づくアーチファクトが生じる。この深さに対する最適値を決定するのは困難である。格子面のソートでは格子の数のほぼ2倍の数のメモリサイズが必要となる。

Anderson ら<sup>7)</sup>は、非構造格子向けボリュームレンダリングを実現するためにポイントベース手法を提案した。彼らはひとつの四面体格子をひとつのポイントで表現し、それらを重ね合わせるによりボリュームレンダリングを実現した。この手法では、1.4M、6.3M個の格子に対して、それぞれ5.3fps、0.3fpsの計算速度を達成したが、ポイントが画素の各軸に沿った四角形以外として投影される場合には描画順序の誤りに基づくアーチファクトが生じる可能性が

ある。

P.Muigg ら<sup>8)</sup>は非構造格子に対するハイブリッドなレイキャスティング法を提案した。彼らの手法は15M個の四面体を22秒で処理した。しかしながらレイキャスティング法はとりわけ大規模なボリュームデータに対しても有効とされているが、格子の隣接情報を含むボリュームデータ全体をメモリに保持しておく必要があり超巨大データに対しては実行不能となる。

Roettger ら<sup>3)</sup>はグラフィックスハードウェアの性能向上が進むとともに格子のソート処理が性能上のボトルネックとなると述べた。彼らは、その時点で、グラフィックスハードウェアや特別なハードウェアにおける性能向上によっても、毎秒1.5M以上の四面体格子を処理できないと予想した。我々はこの予測に賛意を表明し、粒子モデルに回帰することにより非構造格子ボリュームデータを処理する確率統計的なアプローチを提案した。2005年に我々は粒子ベースボリュームレンダリング(PBVR)についての基本アイデアを提案し、いくつかのボリュームデータに対して適用を行った<sup>9)</sup>。第1バージョンでは粒子生成が規則格子に限定されており高品質な画像生成が困難であった。第2バージョンは高精度な粒子モデルを導入し、画質をレイキャスティング法による画像と同等にまで高めた<sup>9)</sup>。第3バージョンは対話性を高めるためにアンサンブル平均による画像生成手法を開発した<sup>9)</sup>。

## 3. 密度発光モデル

Sabellaは、ボリュームレンダリング用に粒子発光モデルと呼ぶレンダリングモデルをはじめて提案した<sup>2)</sup>。このモデルでは、散乱のない不透明な自己発光する粒子がボリュームデータ中に分布するという仮定がおかれた。粒子の大きさは画素の大きさに比べて小さいとされた。このモデルでは、粒子密度は単位体積における粒子の数と決められた。ボリュームレンダリング画像はボリュームデータを通して画素から視点に届く輝度値を画素の色に設定して作成される。ここで、視線上に、中心軸が視線に平行で断面積が $A$ であるような円柱を考えよう。今、視線に沿って $i$ 軸をとり、 $t_0$ と $t_n$ は、 $i$ 軸とボリュームデータとの交点のうち、視点に近い点と遠い点を表現するものとする。粒子密度関数を $\rho(t)$ と表すと、中心が $t$ に置かれ長さが $dt$ であるような微小円筒は、 $A \cdot dt$ の

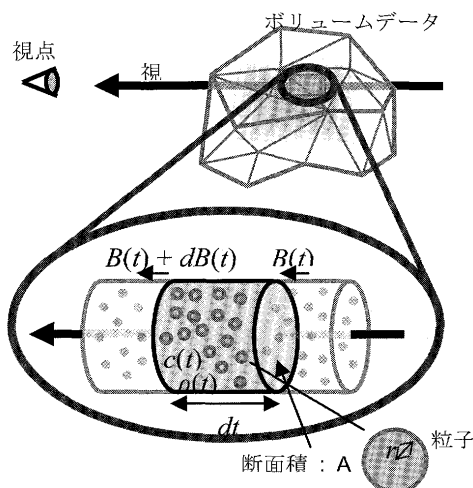


図1 密度発光モデル

体積と、 $N = \rho \cdot A \cdot dt$  個の粒子を持つ。粒子の半径がすべて  $r$  の球であるならその投影面積は  $\pi r^2$  となる。もし、それらがすべて光量  $c(t)$  で拡散的に輝くならば円柱の両底面間の輝度値の差は以下のように記述される (Fig.1)。

$$dB(t) \cdot A = (-B(t) + c(t)) \cdot \pi r^2 \cdot \rho \cdot A \cdot dt \quad (1)$$

従って次のような微分方程式が定式化される：

$$\frac{dB(t)}{dt} = (-B(t) + c(t)) \cdot \pi r^2 \rho(t) \quad (2)$$

式2の両辺に指数項  $\exp(-\int_n \pi r^2 \rho(u) du)$  を乗ずることにより微分方程式は次のように解かれる。

$$B = \int_n^0 c(t) \times \pi r^2 \rho(t) \times \exp(-\int_n^0 \pi r^2 \rho(\lambda) d\lambda) dt \quad (3)$$

ここで、 $B_0 = B(t_0)$  とし、指数項は粒子発光が視点に到達する確率を表す。

式(3)を数値的に計算するために、粒子発光・粒子密度を一定値と見なせる長さ  $\Delta t$  で視線を  $n$  等分し、 $k$  番目における粒子発光・粒子密度を一定値  $c_k \cdot \alpha_k$  とおく。

$$\begin{aligned} B &= \sum_{k=1}^n \int_{t_k}^{t_{k-1}} c_k \pi r^2 \rho(t) \exp(-\int_k^0 \pi r^2 \rho(\lambda) d\lambda) dt \\ &= \sum_{k=1}^n \prod_{i=0}^{k-2} \exp(-\int_{t_{i+1}}^{t_i} \pi r^2 \rho(\lambda) d\lambda) \\ &\quad \times \int_{t_k}^{t_{k-1}} c_k \pi r^2 \rho(t) \exp(-\int_k^{t_{k-1}} \pi r^2 \rho(\lambda) d\lambda) dt \quad (4) \\ &= \sum_{k=1}^n \prod_{i=0}^{k-2} \exp(-\int_{t_{i+1}}^{t_i} \pi r^2 \rho(\lambda) d\lambda) \\ &\quad \times c_k \left(1 - \exp(-\int_k^{t_{k-1}} \pi r^2 \rho(\lambda) d\lambda) \Delta t\right) \end{aligned}$$

$n$  等分された視線をレイセグメントと呼ばれ、 $k$  番目のレイセグメントにおける指数項は不透明度  $\alpha_k$  と定義される。

$$\alpha_k = 1 - \exp(-\int_k^{t_{k-1}} \pi r^2 \rho(\lambda) d\lambda) \quad (5)$$

この不透明度は視線方向に対して粒子発光が遮蔽される確率を表している。レイセグメントの長さは  $\Delta t = t_{k-1} - t_k$  なので式(5)は以下のように簡単化される。

$$\alpha_k = 1 - \exp(-\pi r^2 \rho_k \Delta t) \quad (6)$$

この関係を使って、輝度値は次のように計算される。

$$B = \sum_{i=1}^n c_i \times \left( \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \right) \quad (7)$$

式(7)の計算量がオーダー  $O(n^2)$  であることは明らかである。もし区間  $[t_k, t_n]$  における中間輝度値  $B_k$  を考慮するならば、計算の高速化のために以下にあげる漸化式を導出することができる。

$$B_{k-1} = c_k \alpha_k + (1 - \alpha_k) B_k \quad (8)$$

この方法は、視線に沿って奥から順番(back-to-front)にサンプリング点を評価し、中間輝度値  $B_k$  の  $(1 - \alpha_k)$  をサンプリング点での発光  $c_k$  の  $\alpha_k$  を重ね合わせており、コンピュータグラフィックスの世界ではアルファ合成と呼ばれている。この重ね合わせのためにレイセグメント中心のサンプリング点が視線上でソートされていると仮定すると、この漸化式を用いることにより計算量が  $O(n)$  に減少することがわかる。多くのボリュームレンダリング法では back-to-front 重ね合わせ法を採用している。front-to-back 重ね合わせ法も可能であるが中間不透明度を確保するための記憶容量が必要になる。サンプリング点の視線上ソートは計算速度向上に貢献しているものの大規模非構造格子のボリュームレンダリングを行うときにはボトルネックになってしまう。

#### 4. ポアソン分布と不透明度

確率統計学においてポアソン分布はある決められた時間内で起こる独立事象についてその平均回数が既知のときにその事象が起こる回数の確率を表現する離散的確率分布であると定義されている。ボリュームレンダリングの場合ある決められた空間内で存在する粒子の個数を表すことになる。

ポアソン分布は、レイセグメントで生成された粒子の個数を表す確率変数  $N$  に焦点をあてる。もしそのレイセグメント中に存在する粒子の個数の期待値が  $M$  であった場合、ちょうど  $L$  個 ( $L$  は非負整数すなわち、 $L = 0, 1, 2, \dots$  である) の粒子が存在する確率は

以下のように表される.

$$P(N=L) = \frac{\exp(-M)M^L}{L!} \quad (9)$$

ここで:

- $e$  は自然対数の底 ( $e = 2.71828\dots$ )
- $L$  は体積中に含まれる粒子の数
- $L!$  は  $L$  の階乗
- $M$  は体積中に含まれる粒子の数の平均を表す正の実数

体積中に粒子が存在しない確率は以下のように表現される

$$P(N=0) = \exp(-M) \quad (10)$$

そして, 1 つ以上の粒子が存在する確率は以下のように表現される

$$P(N \geq 1) = 1 - \exp(-M) \quad (11)$$

第  $k$  番目のレイセグメントにおいて存在する粒子の平均個数は以下のように表現される.

$$N_k = \int_{t_k}^{t_{k+1}} \pi r^2 \rho(\lambda) d\lambda \quad (12)$$

式 5 において  $M$  を  $N_k$  で置き換えることにより第  $k$  番目の体積断片において粒子が存在しない確率, すなわち透明度は以下のように表される

$$1 - \alpha_k = P(N=0) = \exp(-N_k) \quad (13)$$

そして第  $k$  番目のレイセグメントにおいて粒子が 1 つ以上存在する確率, すなわち不透明度は以下のように表される

$$\alpha_k = P(N \geq 1) = 1 - \exp(-N_k) \quad (14)$$

式 14 は第  $k$  番目のレイセグメントにおける密度を一定と仮定した場合, 式 6 に一致する.

## 5. アンサンブル平均を使った画像生成

以下の式のように輝度値は各レイセグメントからの発光量についての期待値と見なすことが可能である.

$$B_0 = \sum_{k=1}^n P_k c_k \quad (15)$$

ここで第  $k$  番目の発光量が輝度値に一致する確率は以下のように表される.

$$P_k = \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j) \quad (16)$$

これは最初から第  $(k-1)$  番目のレイセグメントには粒子が存在せず, 第  $k$  番目には 1 以上の粒子が存在す

る確率を表す. この場合, 不透明な発光粒子を仮定している. 輝度値方程式のこの解釈に従えば, ボリュームレンダリングは以下ステップのように確率的粒子生成事象を複数回繰り返すことにより近似することができる.

**Step 1.** 視線上レイセグメント毎にその不透明度に等しい確率でサンプリング点 (粒子または画素) でのレンダリング計算を実行する. 実行されなかった場合は透明として処理される.

**Step 2.** レイセグメント上で視点にもっとも近いサンプリング点の発光量を輝度値 (画素値) とする.

**Step 3.** Step 1 and Step 2 を  $N$  回繰り返し実行し, その結果計算される画像を使って平均画像を作成する.

本手法の有効性を検証するために 1 本の視線上に  $n=100$  のレイセグメントに分割しそれぞれのセグメントの発光量と不透明度にランダムに 0 から 1 までの実数を割り振り, 上記のアルゴリズムを実行し画素値の近似値とする. 正解は式 7 をそのまま計算したものとする. 図 2 は, 繰り返し回数に対する誤差分布を示す. 繰り返し数 100 くらいで十分正解に近くなっていることがわかる.

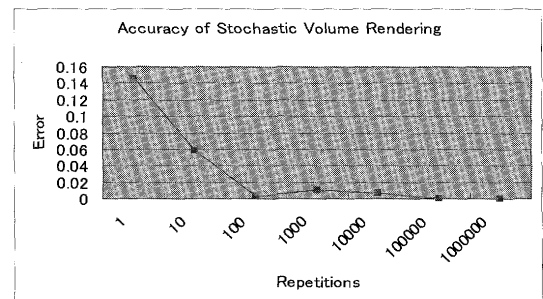


図 2 繰り返し回数に対する誤差分布

このソート不要ボリュームレンダリング法はレイキャスティングによるボリュームレンダリングだけでなく格子投影やスライス重ね合わせ, スプラッティングといった様々なボリュームレンダリング手法にも適用し, ソート不要化とすることが可能である. これらの方法は, 与えられた不規則ボリュームデータを一組のポリゴン形状で近似しそれらを視線からの距離にしたがって, 画像平面上でアルファ合成のために画素展開していく. その際, 各画素で補間計算される不透明度を画素生成確率として利用する. このようなレンダリング計算を複数回繰り返し計算して, 平均化画像を作成することにより実際にアル

ファ合成を行うことなく、生成画像に半透明効果を付与することが可能となる。

この繰り返し画像の平均化処理はアンサンブル平均と見なすことができる。アンサンブルとは概念的に等しい試行実験の仮想的な集合である。ソート不要ボリュームレンダリング法では、試行実験は粒子生成・投影処理を指す。PBVRでは、アンサンブルの各メンバーは粒子モデルとしては同じ属性値（粒子サイズ、粒子密度等）をもち、これらはお互いに独立である。メンバーの数が繰り返し数に一致する。

アンサンブル平均では最終輝度値はすべての繰り返しにおける平均画素値として計算される。つぎにこの平均画素値の揺らぎと繰り返し数の関係について理論的に分析することにする。まず第*i*番目の輝度値について $B^i$ と表記する。そして、アンサンブルの各メンバー同士についてお互いに独立であると仮定すれば、輝度値の統計量（平均・分散）については等しい。すなわち

$$B_{Var} = B^i_{Var} = Var(B^i) \quad (17)$$

最終画素値はすべての繰り返しにおいて平均を取ることにより計算できるので

$$B^{total}(level) = \sum_{i=1}^{level} \frac{B^i}{level} \quad (18)$$

ここで、*level*は繰り返し数を表す。最終画素値の分散は以下のように計算できる

$$\begin{aligned} B_{Var}^{total}(level) &= Var\left(\sum_{i=1}^{level} \frac{B^i}{level}\right) \\ &= \frac{1}{level^2} Var\left(\sum_{i=1}^{level} B^i\right) \\ &= \frac{1}{level^2} \left\{ \sum_{i=1}^{level} Var(B^i) + 2 \sum_{i,j: i < j} Cov(B^i, B^j) \right\} \end{aligned} \quad (19)$$

繰り返し計算における各画素値はお互いに独立であると仮定しているため、各々の分散は等しく、お互いの共分散 $Cov(B^i, B^j)$ についてはゼロと考えることができる。

$$\begin{cases} Cov(B^i, B^j) = 0 \\ V_B = Var(B^i) \end{cases} \quad (20)$$

これより最終画素値の分散は繰り返し数(*level*)の逆数に比例することがわかり、したがってその標準偏差は以下のように評価できる。

$$B_{Dev}^{total}(level) = \sqrt{\frac{V_B}{level}} \quad (21)$$

この式は、標準偏差に閾値が与えられた場合、それを満足する最小繰り返し数が推定できることを示

す。標準偏差に関する閾値が大きくなればなるほど、これを満足する繰り返し数は小さくてすむことがわかる。この繰り返し数は、ボリュームレンダリング計算速度と大きく関係するので、この枠組みを使って、対話的操作時には、繰り返し数を少なく、そうでないときは、繰り返し数を多く設定することにより段階的詳細度制御の実現が可能となる。

## 6. 適用事例

ここではアンサンブル平均を使ったPBVRを紹介する。本手法は筆者らの研究チームが提案している手法をベースにしている。唯一の違いは粒子サイズが画像平面上で一定と見なすかそうでないかということで、ここで紹介する方法は、画像平面上で視点からの距離に従って、粒子の投影像の大きさが変化する。手前のものほど大きく、奥にあるものほど小さく投影され、これは通常のコンピュータグラフィックスで粒子を描画する投影と同じである。

本手法では、我々の従来手法<sup>93)</sup>と同様、まずユーザが設定した伝達関数から導出された粒子密度関数に基づき粒子を生成する。次にこの密度関数に従い、格子単位で粒子を生成する。この格子単位での粒子生成手法では、すべての格子データをメモリに保存しておく必要がなく、大規模データ処理時に有用である。次に、生成された粒子を画像平面に投影し、最も画像平面に近い粒子の投影像だけが有効となるようにして画像生成を行う。これも通常のコンピュータグラフィックスで使われているZバッファアルゴリズムで実現できる。すべての粒子を投影し、作成された画像をアンサンブルの1メンバーとし、同じ粒子密度関数を使って、しかし乱数のシードを変えて生成した粒子を同様に投影し、べつのメンバーに対応する画像を作成する。これを繰り返し行いながら画像を加算し、最後に繰り返し数で除することにより最終画像を得る。

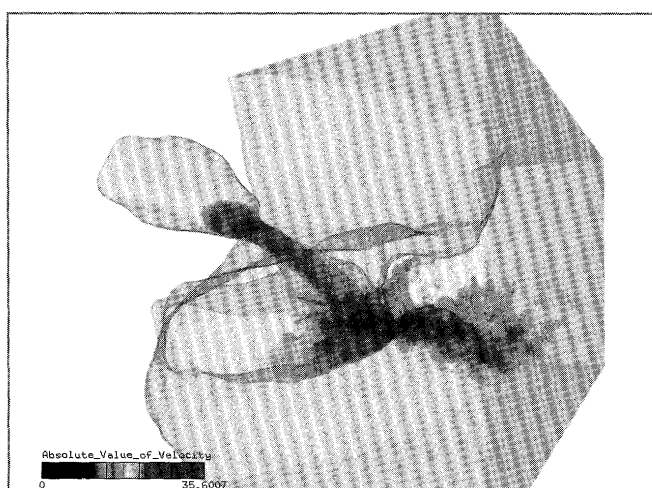
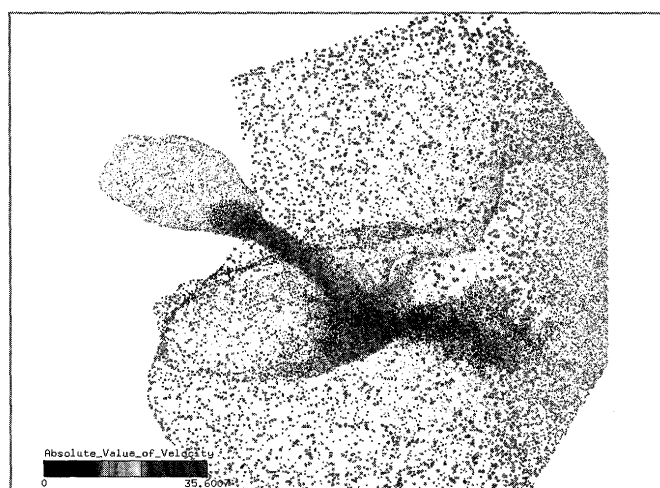
本手法の有効性を検証するために7100万程度の六面体格子からなる”Oral”，そして2400万程度の四面体格子からなる”Pump”の2種類の巨大非構造格子ボリュームデータに適用した。画像生成の計算機環境としては1800MHz AMD Phenom X4 9150 Quad-core CPU, 4.0 GB of RAM, そしてNVIDIA GeForce 9600GT GPU (512MB)を実装したPCを利用した。 ”Oral”データセットを生成したシミュレーションモデルはコーンビームCTによる口腔形状が取得され

数値流体計算用に71,449,236もの六面体格子(六面体1次要素)から構成される3次元メッシュデータが作成された。これらを並列計算機で効率よく計算するために16の領域に分割された。計算はアドベンチャープロジェクトで開発されたソフト<sup>12)</sup>が利用され、可視化対象のシミュレーション結果は16のサブボリュームデータから構成されている。

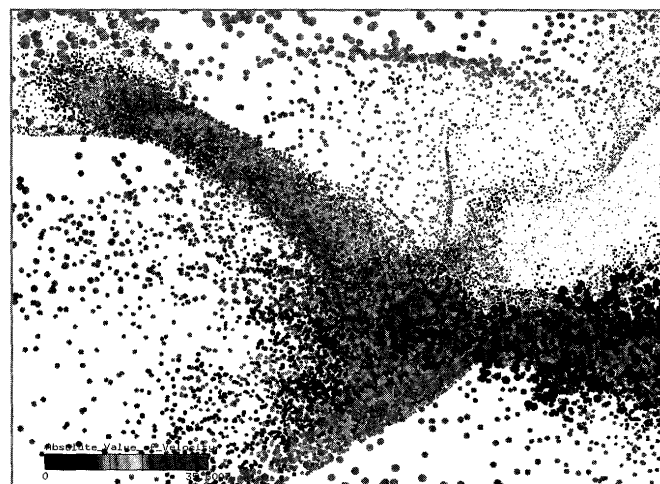
このデータセットのうち速度絶対値に対して粒子ボリュームデータセットを適用した結果が図3である。画像の解像度は800×600で、897万もの粒子データに対し144の繰り返し計算により1枚の画像生成を行いながら、毎秒7.6フレームの高速性を実現している。図3左が繰り返し1の、そして右が繰り返し144の画像である。非構造格子データの可視化の際に重要とされる境界面についても粒子を発生させ半透明で統合表示を行っている。(動画については

[http://www.youtube.com/watch?v=uH\\_FsuGYme0](http://www.youtube.com/watch?v=uH_FsuGYme0) を参照)

もうひとつのデータセット”Pump”は71,449,236ものの四面体格子(四面体2次要素)から構成される。この計算もアドベンチャープロジェクトで開発されたソフト<sup>12)</sup>が利用されており、可視化対象のシミュレーション結果は8つのサブボリュームデータから構成されている。四面体2次要素では、各構成面が必ずしも平面ではない。このために、従来、分散計算結果でよく用いられたサブボリュームごとにボリュームレンダリングを行い、部分画像を作成し、その結果をアルファ合成するという戦略はうまくいかない。なぜなら、各面が曲面となるような格子の場合、視線がそのような面を進入し、またその面から脱出することがあり、部分画像に対して順序を定めることが困難となるからである。このような場合、PBVRは有効である。このデータセットのうちミーゼス応

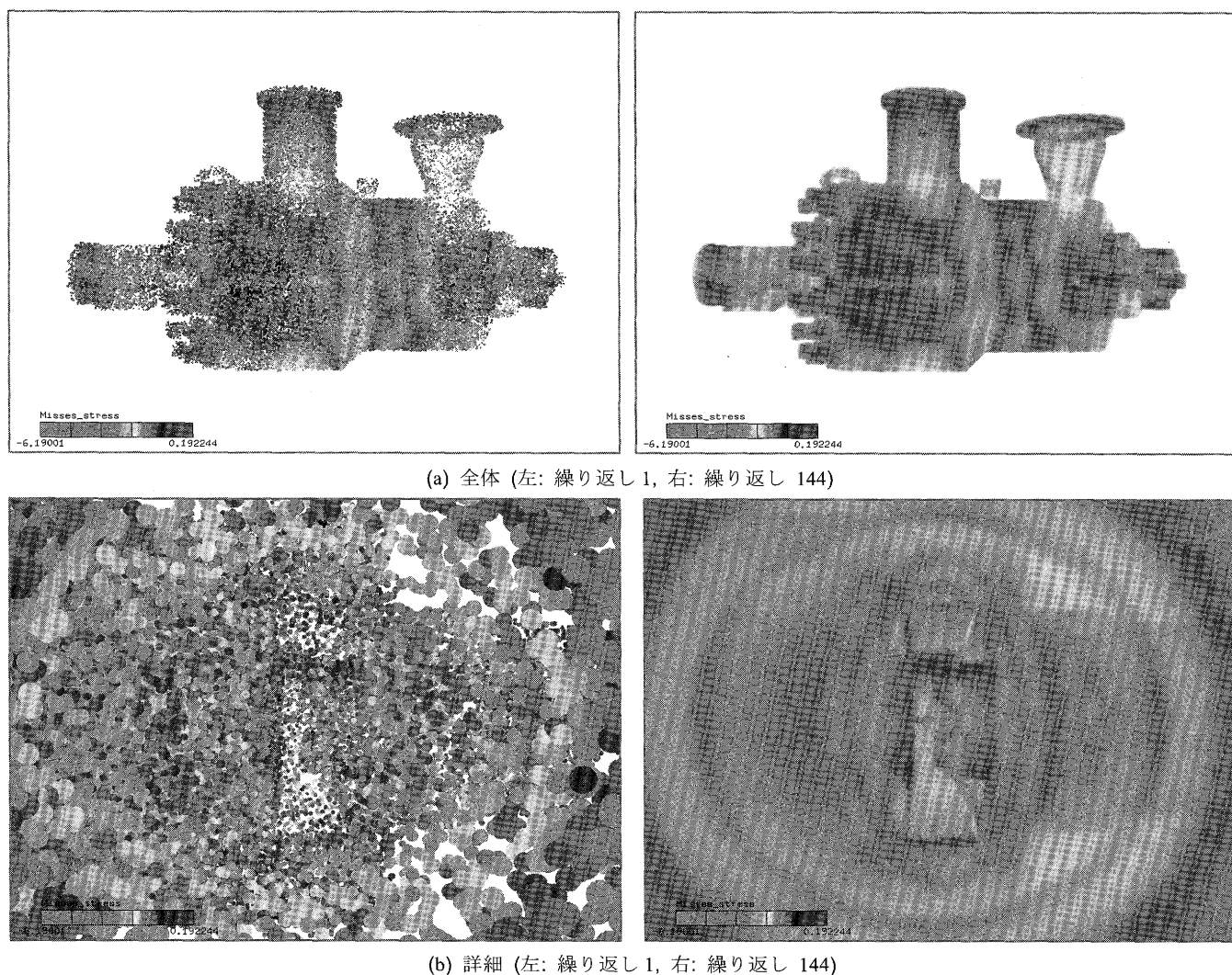


(a) 全体 (左: 繰り返し 1, 右: 繰り返し 144)



(b) 詳細 (左: 繰り返し 1, 右: 繰り返し 144)

図3 口腔シミュレーション結果の可視化 (速度絶対値と境界面) データ提供: 大阪大学野崎一徳講師



(a) 全体 (左: 繰り返し 1, 右: 繰り返し 144)

(b) 詳細 (左: 繰り返し 1, 右: 繰り返し 144)

図 4 ポンプシミュレーション結果の可視化 (ミーゼス応力) データ提供: 東京大学奥田洋二教授

力値に対して粒子ボリュームデータセットを適用した結果が図4である. 画像の解像度は $800 \times 600$ で, 1585万もの粒子データに対し144の繰り返し計算をして1枚の画像生成を行い, 毎秒5.2フレームの高速性を実現している.(動画については

[http://www.youtube.com/watch?v=uH\\_FsuGYme0](http://www.youtube.com/watch?v=uH_FsuGYme0) を参照)

## 7. まとめ

本解説では, ユーザの設定した伝達関数に従って生成された粒子から構成される粒子モデルに回帰することにより非構造格子ボリュームデータに対して効率よくボリュームレンダリングする手法について説明を行った. 非構造格子ボリュームデータにおいて性能上問題となるソート処理についてアンサンブル平均を利用した画像生成を取り入れることにより高品質な画像生成が可能となることを説明した. 統

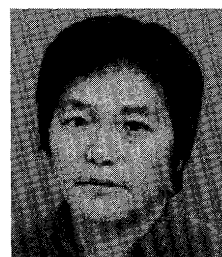
計確率統計学的の知見を使うことによって, アンサンブル平均における繰り返し数を増やすと画素値におけるゆらぎが減少することを示した.

## 参考文献

- 1) B. McCormik, T. DeFanti and M. Brown, Visualization in Scientific Computing, National Science Foundation, 1987.
- 2) P. Sabella, A Rendering Algorithm for Visualizing 3D Scalar Field, Computer Graphics, Vol. 22, No. 4, pp. 51-58, 1988.
- 3) S. Roettger, and T. Ertl, Cell Projection of Convex Polyhedra, In Proc. of Volume Graphics 2003, pp. 103-107, 2003.
- 4) B. Csébfalvi, and L. Szirmay-Kalos, Monte Carlo Volume Rendering, In Proc. of IEEE Visualization 2003, pp. 449-456, 2003.

- 5) B. Csébfalvi, Interactive Transfer Function Control for Monte Carlo Volume Rendering, In Proc. of IEEE Symposium on Volume Visualization and Graphics 2004, pp. 33-38, 2004.
- 6) S. P. Callahan, M. Ikits, J. L. D. Comba, and C. T. Silva, "Hardware-Assisted Visibility Sorting for Unstructured Volume Rendering," IEEE Transactions on Visualization and Computer Graphics; May/Jun 2005.
- 7) E. W. Anderson, S. P. Callahan, C. E. Scheidegger, J. Schreiner, and C. T. Silva, "Hardware-Assisted Point-Based Volume Rendering of Tetrahedral Meshes," Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), 2007.
- 8) P. Muigg, M. Hadwiger, H. Doleisch, H. Hauser, "Scalable Hybrid Unstructured and Structured Grid Raycasting," IEEE TVCG, 13(6): 1592-1599, 2007.
- 9) N. Sakamoto, and K. Koyamada, "Particle Generation from User-specified Transfer Function for Point-based Volume Rendering," IEEE Visualization Proceedings Compendium, pp.125-126, 2005.
- 10) K. Koyamada, N. Sakamoto, S. Tanaka, "A Particle Modeling for Rendering Irregular Volumes," International Conference on Computer Modeling and Simulation (UKSIM 2008), pp.372-377, 2008
- 11) Z. Ding, N. Sakamoto, T. Kawamura and K. Koyamada, " GPU-based Opaque Splatting for Unstructured Volume Rendering," IEEE Pacific Visualization Proceedings Compendium, 2009(to be presented).
- 12) ADVENTURE Project,  
<http://adventure.sys.t.u-tokyo.ac.jp/project/>

(こやまだ こうじ／京都大学)



小山田 耕二

1985 年 3 月京都大学大学院修士課程修了. 1985 年 4 月日本アイビーエムを経て, 2003 年 8 月から現職. 大規模シミュレーション計算結果の可視化技術・最適化技術の研究に従事. 工学博士. 日本シミュレーション学会理事, 日本可視化情報学会理事, システム制御情報学会理事