タイルド表示装置を用いた 大規模可視化の可能性

小山田耕二○ (京都大学)，坂本尚久(京都大学)

High-resolution Visualization System of Large-scale irregular Volumes on a Tiled Display Wall

Koji KOYAMADA, Naohisa Sakamoto

ABSTRACT

In this paper, we propose a system that visualizes a large-scale irregular volume dataset on a large-scale display with the Particle-Based Volume Rendering (PBVR) technique and the Scalable Adaptive Graphics Environment (SAGE). Medical imaging devices such as a Computed Tomography (CT) system can easily model human internal organs in 3-D geometries, and a computational fluid dynamics system can then calculate flow fields from these geometries. A collaborative visualization system using the Tiled Display Wall (TDW) system uses the simulation result. We applied the proposed system to the calculated oral cavity flow field to evaluate its effectiveness.

Keywords: Tiled display wall, large-scale volume rendering

1．緒　論

計算機と有限要素法ソフトウエアの性能向上に伴い，可視化ソフトウエアの役割が重要になってきた．何千万もの要素からなるモデルが作成可能となったもののその計算結果が大規模で従来の可視化ソフトウエアでは表示が困難となってきている．大規模計算ではモデルを領域分割したあと分散メモリ型計算機を使って計算することが多く，結果として可視化の対象はひとつにまとめるのが困難であるような複数の領域ごとの非構造格子データが生成される．本解説では可視化技術としてボリュームレンダリング法を考える．

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のある重要な技術として位置づけられている．Sabella[1]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粒子ボリュームレンダリング法[2]を説明する．

可視化は，研究者に物理現象理解のきっかけを与えるものである．その可視化の効能は，可視化画像の解像度の高さと関係があると認識されている．人間の網膜に適う解像度は 3.6 億画素であり，通常の表示装置の解像度数百万画素とは 2 桁程度の違いがある．この違いを克服すべくタイルド表示装置を用いた高解像度環境の構築が世界的に進められている．現在，テキサス大学オースチン校が 75 枚の表示装置を使って，約 3 億画素程度の解像度を実現しており，これが世界最大とされている．我々は．複数画面向け高解像度画像表示ライブラリ OpenCABIN[3]を利用して 40 枚のタイルド表示装置を構築し，その上で数千万要素の大規模費構造格子データの実時間ボリュームレンダリング表示環境を実現した．まず，粒子ボリュームレンダリング法・OpenCABIN の概要について述べた後，開発システムと適用事例について説明する．

2．粒子ボリュームレンダリング法

2．1　輝度値方程式

ボリュームレンダリングにおいて輝度値Bは次のように計算される．

$$B = \sum_{k=1}^{n} c_k \times \left(\alpha_k \prod_{j=1}^{k-1} (1-\alpha_j) \right) \qquad (1)$$

式 1 では，粒子発光・粒子密度を一定値と見なせる長さ Δt のレイセグメントで視線を n 等分し，k 番目における粒子発光・不透明度・粒子密度を一定値 c_k ・α_k・ρ_k とおく．ここで不透明度は粒子密度，粒子径(r)，レイセグメント長さ Δt を用いて以下のようにあらわされる．

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

式 1 の計算においては back-to-front 重ね合わせ法を採用している．この重ねあわせ法を効率よく行うにはサンプリング点のソート処理が必要となるが，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，このソート処理そのものがボトルネックになってしまう．次に不透明度を確立統計的な立場で解釈し，ソート処理を必要としないボリュームレンダリング手法について説明する．

2．2　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， $L = 0, 1, 2, \ldots$ である)の粒子が存在する確率は以下のように表される．

$$P(N = L) = \frac{\exp(-M) M^L}{L!} \quad (3)$$

ここで:

- e は自然対数の底 ($e = 2.71828\ldots$)
- 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

$$P(N = 0) = \exp(-M) \quad (4)$$

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

$$P(N \geq 1) = 1 - \exp(-M) \quad (5)$$

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．

$$N_k = \int_{t_k}^{t_{k-1}} \pi r^2 \rho(\lambda) d\lambda \quad (6)$$

式 5 において M を N_k で置き換えることにより第 k 番目のレイセグメント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．

$$1 - \alpha_k = P(N = 0) = \exp(-N_k) \quad (7)$$

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．

$$\alpha_k = P(N \geq 1) = 1 - \exp(-N_k) \quad (8)$$

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 8 は式 2 に一致する．

2．3　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．

$$B = \sum_{k=1}^{n} c_k \times P_k \quad (9)$$

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．

$$P_k = \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j) \quad (10)$$

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かえ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 と 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 本の視線上を 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．Fig. 1 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．

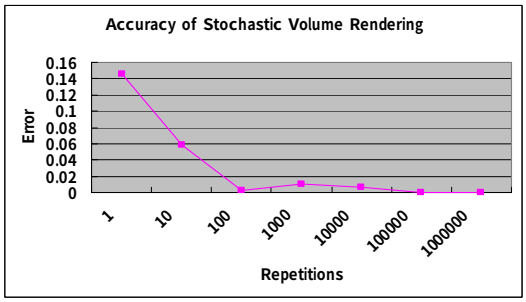

Fig. 1:Error Ddstribution with respect to repetitions

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

$$B_{Var} = B^i{}_{Var} = Var(B^i) \tag{11}$$

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

$$B^{total}(level) = \sum_{i=1}^{level} \frac{B^i}{level} \tag{12}$$

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．

$$\begin{aligned} & B^{total}_{Var.}(level) \\ &= Var(\textstyle\sum_{i=1}^{level} \frac{B^i}{level}) \\ &= \frac{1}{level^2} Var(\textstyle\sum_{i=1}^{level} B^i) \\ &= \frac{1}{level^2} \left\{ \textstyle\sum_{i=1}^{level} Var(B^i) + 2\sum_{i,j;i<j} Cov(B^i, B^j) \right\} \end{aligned} \tag{13}$$

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ので，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．

$$\begin{cases} Cov(B^i, B^j) = 0 \\ V_B = Var(B^i) \end{cases} \tag{14}$$

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．

$$B^{total}_{Dev.}(level) = \sqrt{\frac{V_B}{level}} \tag{15}$$

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．

3．OpenCABIN を使った分散 PBVR

3．1　OpenCABIN 概要

OpenCABIN は没入型ディスプレイやタイルドディスプレイに代表される PC クラス型のマルチスクリーンシステムに対応した基盤ライブラリである．OpenCABIN では，描画機能に関して，OpenGL だけでなくその他の既存グラフィックスライブラリに柔軟に対応するために，描画機能部とその他の機能部とを独立して管理する Master/Renderer 機構を特徴とし，スクリーンごとに定義される Renderer は，フレーム間同期をとることなく，それぞれが非同期的に動作する．この仕組みにより，Renderer 間の描画負荷の不均衡（load imbalance）から生じる描画のオーバヘッドを低減し，効率よく空間全体を描画することができる．しかし，各 Renderer の表示空間は表現する空間の一部または同一空間であるため，スクリーンごとの処理時間の差によるパラレルワールドの発生を防ぐためには，時間的に変化する変数などは，各 Renderer 間で共有化する必要がある．OpenCABINLib では，このような変数を，マクロ関数

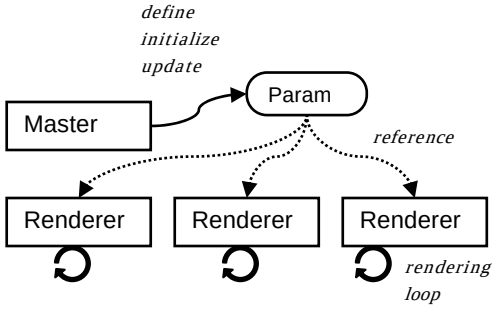

Fig.2: Synchronization mechanism on OpenCABIN.

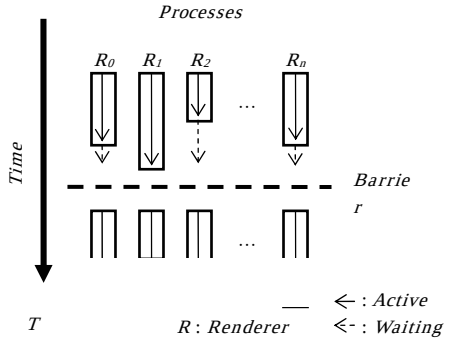

Fig.3: Barrier synchronization.

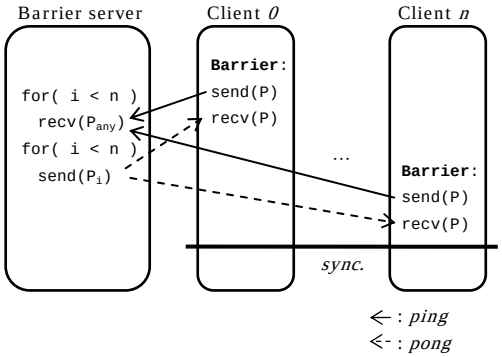

Fig.4: ping-pong process.

を利用して定義/初期化/更新/参照を行うことにより，ユーザは簡単に変数の共有化を行うことができる(Fig.2).

3．2　バリア同期

マルチスクリーン全体で描画される空間全体を実時間で効率よく描画するためには，フレーム間で同期をとらない非同期型レンダリング方式は効果的である.しかし，PC（ディスプレイ）間で描画負荷の大きく異なるレンダリングの場合は，フレームの飛びが多発する可能性があり，結果として部分的に情報が欠落した映像が表示される可能性がある．本論文では，フレーム間で同期をとるためのバリア同期機構（Fig.3）を実装する.

Fig.4 にサーバークライアント間で ping-pong 処理によるバリア同期の仕組みを示す．最初に，同期サーバを起動しクライアントからのメッセージ受信待ち状態にする．次に，クライアントを起動しサーバに対してメッセージを送信（ping）する．そして，クライアントはメッセージ送信後，ただちにメッセージ受信状態にする．マスタは，全クライアントからのメッセージを受信後，それらのクライアントに対して，返信メッセージを送信（pong）する．クライアントは，サーバから返信メッセージを受信後,以降の処理を実行する.このようにして，クライアントはサーバとの ping-pong 処理をおこなうことにより，同期をとることができる.

3．2　分散 PBVR

ここでは大規模非構造格子データを処理する上で有用な分散 PBVR について説明する.効率のよい処理のために我々は，与えられた伝達関数から粒子密度を推定することによって格子毎に粒子群を発生する手法を開発した．それぞれの格子は任意順序で，投影対象となる粒子を生成することができる．これにより分散環境で効率よく稼働させることが可能となる.

我々はPCクラスタを使って分散PBVRを実装した(Fig. 5 参照). 本システムは複数の処理ノードと１台の制御ノードから構成される．これらのノードはギガビットイーサネットを通じてネットワーク接続されている．本システムは７つの処理ステージ(A. ファイルロード，B. 粒子生成，C. 粒子転送，D. 粒子レンダリング，E. ビューポートカリング，F. TDW での表示）から構成される．各ステージで処理の説明は以下の通りである.

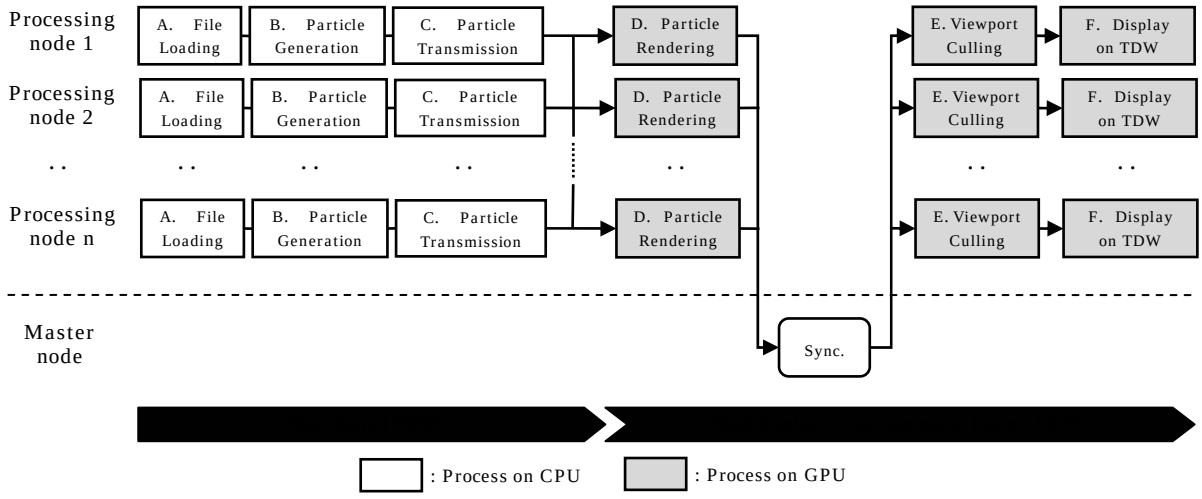

Fig 5 分散 PBVR システム構成.

A.　ファイルロード

本システムでは，非構造格子データは格子接続情報，格子点座標，スカラデータ情報から構成される．処理ノードにおいて複数のデータファイルとして表現され，処理ノードの CPU メモリにロードされる．

B.　粒子生成

処理ノードにおいて粒子密度は格子毎に推定される．粒子は推定された密度場にしたがって処理ノードの CPU メモリにおいて生成される．

C.　粒子転送

生成した粒子を自ノード以外の全処理ノードに転送する．転送中，粒子は座標値，勾配ベクタ，スカラ値，識別子として表現される．

D.　粒子レンダリング

他の処理ノードから転送された粒子を受信し，任意の順序により GPU 上でレンダリングされる．このとき，制御ノードと処理ノード間で ping-pong 処理を行うことによってフレーム間同期をとる．

E.　ビューポートカリング

担当するディスプレイに応じたビューポートでカリング処理を行う．

F.　TDW 表示

担当領域ごとにフレームバッファの映像を TDW 表示装置に出力する．

本システムにおいてタイルド表示装置向けの処理としては，処理ステージ EF が該当する．Fig. 6 にタイルド表示装置を使った分散 PBVR システム概要を示す．

Fig. 6 タイルド表示装置を使った分散 PBVR システム概要

４．適用事例

本システムの有効性を検証するためにアドベンチャープロジェクトで開発された有限要素法ソフトウェアによって生成された大規模非構造格子データに本システムを適用した．本システムは，1 台の制御ノードと 10 台の処理ノードからなる PC クラスタシステム，および，1 台の処理ノードにつき 4 面の LCD ディスプレイが接続された 40 面（8x5）のタイルド表示装置から構成される．PC クラスタシステムを構成する計算機環境としては Intel Core 2 Duo 2.6GHz CPU，1．0 GB RAM，そして NVIDIA Quadro 570 GPU (256MB)を実装した PC を利用した．大規模非構造格子データとしては，2400 万程度の四面体格子からなる”Pump”の巨大非構造格子ボリュームデータを利用した．このデータは並列計算機における計算結果であるために複数のデータファイルから構成される．

”Pump”は 26，289，770 もの四面体格子（四面体２次要素）から構成される．この計算にはアドベンチャープロジェクトで開発されたソフト [4]が利用されおり，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，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従来分散ボリュームレンダリングでよく用いられた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不可能となるからである．このような場合，本手法は有効である．”Pump”データセットのうちミーゼス応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 Fig. 6 である．画像の解像度は 10,240×5,120 で，9,280 万もの粒子データに対し 16 の繰り返し計算をして 1 枚の画像生成を行い，回転操作時には繰り返し回数を 1（580 万粒子）にして描画する詳細度制御を行うことで毎秒数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=E2Hkk4TJ2Rc を参照)

参 考 文 献

1) P. Sabella, A Rendering Algorithm for Visualizing 3D Scalar Field, Computer Graphics, Vol. 22, No. 4, pp. 51-58, 1988.
2) 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
3) OpenCABIN library, http://sourceforge.net/projects/opencabin/
4) ADVENTURE Project, http://adventure.sys.t.u-tokyo.ac.jp/project/