

非構造格子のデータを対象としたボリュームレンダリング

7P-9

西松 研 河野 洋一 福盛 秀雄 村岡 洋一
早稲田大学

1 はじめに

計算機の高速化により、大規模な数値シミュレーションが行なわれるようになり、数値結果を可視化する技術も重要視されるようになってきた。その中でも、データ全体を可視化対象とするボリュームレンダリング法が脚光をあびているが、計算時間が非常にかかることが問題とされている。

そこで、処理時間の短縮を目指し、並列計算機を利用した高速化の研究がおこなわれてきた。その対象となったのは、扱いやすさから立方体のような構造格子データだった。しかし、有限要素法のように四面体のような非構造格子を扱うものも存在する。非構造格子からサンプリングする際、内挿は、そのサンプリング点を含む四面体格子で行なわれるのが望ましい。そこで、非構造格子のデータ構造を利用して可視化する手法、さらには高速化の手法が必要とされる。

本研究の目的は、有限要素法のような非構造格子を扱う数値シミュレーション結果をダイレクトに実時間可視化するシステムの構築である。

本稿では、四面体を格子構造としたボリュームレンダリング法を実装したので、報告する。

2 理想的な計算

非構造格子を対象とした計算結果のボリュームレンダリング実時間可視化を目指す上で、並列計算機の利用を考える前に、レンダリング処理自体の効率化について考える。

ボリュームデータをダイレクトに可視化する手法として、volume ray casting 法や splatting 法がある。

volume ray casting 法は、ユーザの視点からピクセルを通る光線を追跡していき、そのピクセルに影響するボリュームデータをサンプリングしていく。サンプリングの途中で、計算されたピクセル値が不透明とみなす基準に達した時に計算を打ち切ることができる。この方法では同一ボリュームへのアクセスが繰り返し発生する。さらに、透明とみなされたボリュームは、アクセスされても結果に影響を及ぼさないため、そのボリュームを読み飛ばす仕組みが必要になる。

splatting 法は、ボリュームデータからカーネルを作成し、スクリーンに投影する。その結果をスクリーン側で混ぜ合わせる手法である。ボリュームデータ単位で計

算するため、ボリュームデータ数の回数の計算ですむ。しかし、最終映像に関係ないデータも計算対象となるため、無駄な計算が発生してしまう。

そこで、ボリューム単位で計算していくが、透明とみなされたボリュームデータにはアクセスせず、最終映像に影響をもたないボリュームデータの計算も行わない方法が理想である。

3 可視化手法

Philippe ら [1] [2] は、構造格子を対象とし、内挿を3次元格子の問題から2次元格子の問題に置き換え、先の章であげた両者の手法の長所をおり混ぜることで、従来に考案された方法よりも処理時間が短い手法を提案した。内挿時の次元の落とし方は以下の方法による。

ボリュームデータから基本軸(下の図ではz軸)に垂直な複数のスライス抽出し、基本軸正面から見た座標系に変換する。この変換によって、3次元ボリュームデータは複数の2次元スライスに変換される。

スライス上の情報を用いて、前面のスライスからスクリーンに投影していく。最後にスクリーンを元の座標系に戻すという方式である。

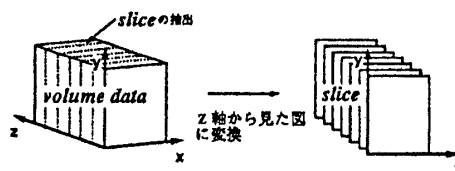


図1: Philippe らの手法

そこで、四面体のような非構造格子にもその手法を適用することを試みた。

3.1 適用上の問題点

四面体データのような非構造格子データは点情報(座標・物理量)と四面体を構成する点番号情報のみからなると仮定する。

四面体データに同手法を適用するためには、スライス上のスキャンライン単位でボリュームデータを読んでいく仕組みが必要となる。同時に、スライス上のサンプリング点を投影していくとき、そのサンプリング点の物理量を求めるためには、どの四面体内の点か知る必要がある。

そこで、四面体の面は他のどの四面体の面を共有しているのか、あるいはボリュームデータ空間を、あるスラ

イス上で切断したときにできる三角形の辺は他のどの三角形の辺と共有しているかを格納するテーブルを作成した。

テーブルを参照することで、スライスで切断した面におけるスキャンライン単位のデータアクセスが実現可能となった。

3.2 テーブルの作成手法

テーブルの作成を下記の手法で行なう。

1. 非構造格子データに対し、基本となる x, y, z 軸方向に垂直なスライスをそれぞれ作り、その面と四面体が交差する面を求める。交差によってできた面をそのスライス上の情報として持たせる。交差によってできた面が四角形の場合は2つの三角形に分割することで、全て三角形のデータとする。
2. スライス上の三角形に対し、その三角形の辺の共有情報を作成する。
3. 三角形の辺の共有情報をもとに、スキャンライン単位にどの三角形を通るかを調べる。

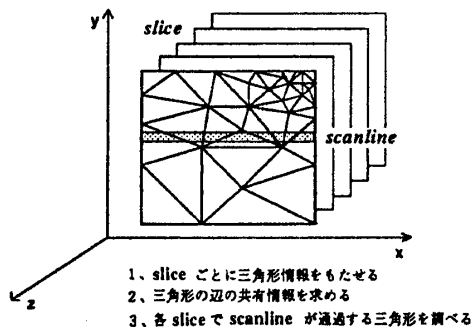


図 2: テーブルの作成

3.3 実験

1*1*1 の 3次元空間上でポロノイ分割 [3] された 700 節点、4378 個の四面体格子データに対し、(0.5, 0.5, 0.5) の位置に一番大きいポテンシャル値を与え、その位置から遠ざかるにつれ、1次関数的に減少するポテンシャル値を与えることで、逐次計算機上で可視化実験を行ない、動作確認をした。

4 本手法の考察

今後、非構造格子データのボリュームレンダリング法を実時間でおこなう手段として、並列計算機の利用を考えている。そこで、本手法の利点と問題点を考える。

4.1 利点

- 一度、前処理を行えば、テーブル情報をもとにレンダリングをおこなうことができる。
- 2次元スライス上に四面体格子の情報を反映させて、三角形の情報に落しているため、もとの四面体で内挿したのと同じ精度が得られる。つまり、

四面体内での内挿から三角形内での内挿に次元を落すことができるため、計算量を減らすことができる。

- ボリュームデータでは非構造格子の形状から、ボリュームが透明とみなされる部分とそうでない部分の切り分けが難しいが、スライス上では、容易。

4.2 問題点

- 最初に与えられた情報に対し付属的な情報が増えるため、メモリを消費する。
- 前処理を施すため、ユーザが可視化対象とするデータの範囲を変更した場合、あるいは格子分割の情報が変化した場合は、テーブルを作り直す必要がある。
- 四面体一つの情報が入り、スライスやスキャンラインによって、幾つにも分断されてしまうため、ボリュームデータ単位の処理とは言えない。

4.3 並列化に向けて

並列計算機の各 PE への効率的なデータ配置を考える。そのとき、四面体格子の形状よりは、3次元ボリュームデータから変換した複数の2次元スライスやスライス上の三角形、スキャンライン単位の方が扱いやすく、その単位は負荷分散する上での仕事の単位にもなる。

また、スライス情報に落すことで、Philippe ら [1] [2] のアルゴリズムによる理想的な計算に近い考えを利用することができる。

現在はボリューム全体に対し、一定間隔おきのスライスを作成している。しかし、格子情報は密な部分と疎な部分があるため、ボリュームデータ全体に対し、スライスを作成するのではなく、格子の状態に応じた部分的なスライスの作成も考えられる。

5 おわりに

Philippe ら [1] [2] の開発したアルゴリズムを非構造格子を対象としたボリュームレンダリング法に適用する手法について述べた。

今後は、実際の非構造格子を用いた数値シミュレーション結果の可視化に使用できるよう、大きなデータを扱いテストすることや、並列計算機を利用して高速化し、実時間可視化を目指す。

参考文献

- [1] Philippe G. Lacroute : FAST VOLUME RENDERING USING A SHEAR-WARP FACTORIZATION OF THE VIEWING TRANSFORMATION, *Technical Report: CSL-TR-95-678*, September, 1995
- [2] Philippe Lacroute, Marc Levoy : Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation, *ACM SIGGRAPH 94*, pp.451-458, July, 1994
- [3] 河野洋一他 : 有限要素法要素分割の並列化 -ポロノイ分割の並列化-, *ハイパフォーマンスコンピューティング*, 60-8, pp.43-48, 1996