

# 大規模数値シミュレーションの可視化のためのデータ階層化

吉田 有宏 杉原 光太 武井 利文<sup>†</sup>

数値シミュレーションの大規模化に伴い、その莫大な計算結果の可視化に長時間を要するようになった。データ階層化を行えば、画素より細かな詳細情報や画像の範囲外の情報といった不要な情報を極力アクセスすることなしに、可視化に必要な情報のみを含んだ数値データをすぐに得ることができるため、大規模データであっても高速な可視化を実現できる。本研究では最も汎用性の高いデータ形式である非構造四面体格子データの階層化を実現した。さらに、階層化データの可視化確認と、その処理時間の評価を行った。

## Hierarchical Representation of Large-Scale Simulation Results for High-Speed Visualization

Arihiro Yoshida, Kouta Sugihara, and Toshifumi Takei<sup>†</sup>

With the expansion of the scale of numerical simulations, the time to visualize the huge computation results has become unacceptably long. By using hierarchal representation, summary data almost only including information required for visualization can be obtained soon, without accessing as much of invisible information as possible, such as detailed information finer than a pixel, or information out of view. Therefore, hierarchal representation realizes high-speed visualization even of large-scale of data. We developed the method to create hierarchal representation of data on tetrahedral unstructured meshes, which are the most generic of all types of meshes. Furthermore, we tested visualization of data in hierarchal representation, and evaluated its processing time.

### 1. はじめに

近年、PC クラスターの普及、地球シミュレータ、米国 ASCI 計画などで代表されるように、過去考えられなかったような大規模、超大規模の数値シミュレーションが現実のものとなりつつある。一方で、その膨大な数値データの可視化に時間がかかり、結果の解釈が円滑に行えなくなってきた。例えば地球シミュレータの場合、総メモリ容量は10TB であり、これほどのオーダーの巨大データの場合、従来どおりの可視化によっては、結果確認が非常に困難となる。しかし、数値データがどんなに膨大でも、可視化画像に表れるのは全データのほんの一部かつ一定量の情報

であり、ほとんどの情報は可視化に不要である。データ階層化を行えば、可視化に必要な十分な情報を効率よく取得できる。そのため、可視化範囲やデータ規模によらない高速な可視化が品質を損なうことなく実現できるはずである。本研究では三次元非構造格子データの階層化を実現し、大規模データに対する有効性を検証した。

### 2. データ階層化とは

まず、可視化に不要な情報とは何かを考えてみる。図 1 に示すように、数値データ全体を可視化すれば細部は見えなくなり、細部を可視化すれば全体のほとんどが視界から外れて見えなくなる。画素より細かな詳細情報や画像の範囲外の情報は画像には表れないから、可視化範囲によらず一部の情報だけで

<sup>†</sup> NEC インターネットシステム研究所  
Internet Systems Research Laboratories, NEC Corporation

十分な可視化を行うことができる。しかも、数値データがいくら膨大でも、可視化画像の画素数は一定であるから、一度に可視な最大情報量は不変である。従って、可視化に必要な情報のみを少ない処理量で抽出する方法があれば、数値データの規模によらず、ほぼ一定の処理コストで高速な可視化を行えるはずである。

そこで、図 2 のように数値データを単純化し、各詳細度の概要データを階層的に保持することを考える。これにより、任意箇所における任意詳細度の数値データをすぐに取得でき、不可視な情報に極力アクセスすることなしに可視化が行える。これをデータ階層化と呼ぶ。実際には、各階層間の差分に相当する詳細化データと、最も詳細度の低い単純化データのみを保持し、階層数の増加に伴うデータサイズの増加を抑える。図 3 に示すように、元データや概要データから一つ低い詳細度の概要データと詳細化データを生成する操作を分解と呼ぶ。逆に元データや各詳細度の概要データは、一つ高い詳細度の概要データと詳細化データにより再構成することができる。詳細なデータほど何段も再構成を行う必要があるが、再構成は局所的に行うことができるので、最小限のデータで必要箇所のみの詳細データを再構成できる。

このように、データ階層化は、大規模データであっても、品質を損なわずに全体や各詳細度の細部をほぼ一定コストで可視化できるため、大規模数値データに必須の機能となる。

### 3. データ階層化の利用

データ階層化を数値シミュレーション結果の可視化に利用する場合、システム構成は次のようになる。

図 4 は階層化データ生成システムの例である。保存された計算結果から階層化データを生成する構成(a)と、計算結果が得られると同時に階層化データを直接生成する構成(b)がある。いずれの場合も、分解処理は入出力共に大規模なデータであるため、処理能力が高く、記憶容量が大きい計算サーバまたはデータサーバで階層化データの生成を行うことになる。

図 5 は階層化データ可視化システムの例である。ここに示したいずれの場合も、処理の流れはまったく

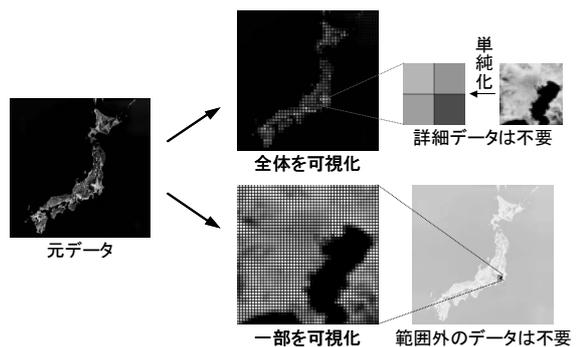


図 1 可視化に必要な情報

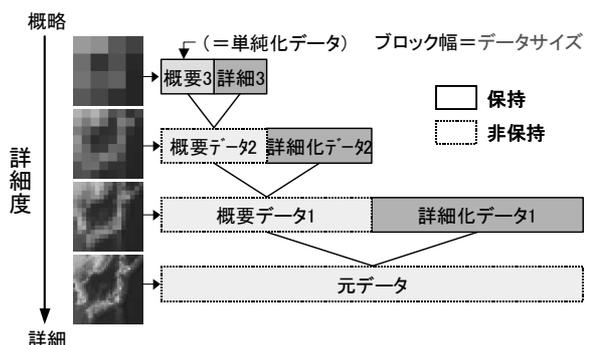


図 2 階層化データの基本構造

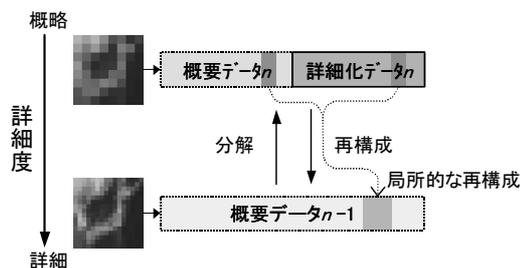


図 3 分解と再構成

同じである。ユーザインタフェース(UI)は、ユーザの要求に応じて可視化パラメータを生成し、これを可視化モジュールに送る。可視化パラメータとは、どのデータのどの部分をどの向きに可視化するかといった情報である。可視化モジュールは、この情報をもとに再構成パラメータを生成し、これを再構成モジュールに送る。再構成パラメータとは、どのデータのどの部分をどれくらい詳細に再構成するかといった情報である。再構成モジュールは、この情報をもとに階層化データから格子データを再構成し、これを可視化モジュールに送る。可視化モジュールは、この再構成格子データを通常とまったく同様の可視化処理を行い、得られた画像を UI に送る。UI はこの画像を端末画面に表示する。

図 5 にいくつかのシステム構成を示しているが、こ

これらの違いは、再構成格子データの可視化をどこで行うかにある。可視化サーバで行う構成(a)と、計算サーバまたはデータサーバで行う構成(b)と、端末で行う構成(c)がある。専用ハードウェアを搭載した高速な可視化サーバがあり、可視化サーバと計算/データサーバ間のネットワークが、格子データを短時間に転送するに足る十分な帯域を持つ場合、構成(a)が一番有効である。この贅沢な構成と対照的に、一番経済的なのは構成(b)である。ネットワークは画像を遅滞なく転送できる程度の帯域を持っていればよく、可視化サーバを別途用意する必要もない。これらとは少し趣が異なるが、得られた結果の格子データの興味ある一部、もしくは単純化された全体を、ローカル端末上にダウンロードして、じっくり可視化したいという場合は構成(c)がよい。

図 6は階層化データ経由の実時間可視化システムの一例である。これは図 4の構成(b)と図 5の構成(a)を合体したものである。シミュレーションプログラムから時刻ステップ毎に出力される計算結果から直接使い捨ての階層化データを生成し、これを用いて可視化を行うというものである。

#### 4. データ階層化の方式

一口にデータ階層化とは言っても、様々な方式が提案されている。最も基本的な階層化方式は構造格子データに対するもので、単純化操作として単純間引きや平均化を用いたものである。我々は、構造格子データのデータ縮小化と称して、単純間引きによる高速なデータ階層化の研究も行っている。これらの発展形にあたるのがウェーブレットを用いる方式[1],[2]であり、数学的明快さと応用性ゆえに、画像やCTボリュームデータによく適用されている。

非構造格子データにも適用できる方式としては、ウェーブレットライクな方式[3],[4]、octreeを用いる方式[5],[6]、幾何単純化による方式[7]-[13]がある。ウェーブレットライクな方式は、構造的な格子に再サンプリング(リメッシュ)する必要があり、その処理コストと容量コスト、リメッシュのために非可逆となる点で、大規模数値データに適用するには問題がある。octreeを用いる方式も、構造的なセルに再サンプリングする必要があり、非可逆な階層化である。幾何単純化による方式だけが、非構造格子データを非

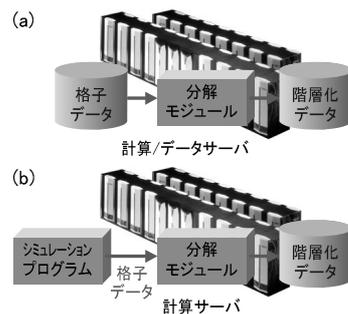


図 4 階層化データ生成システム

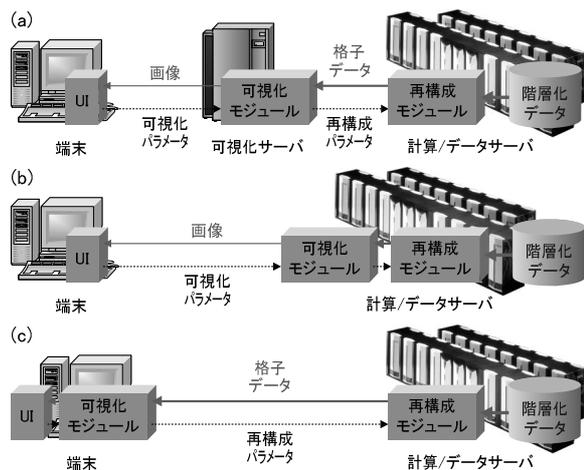


図 5 階層化データ可視化システム

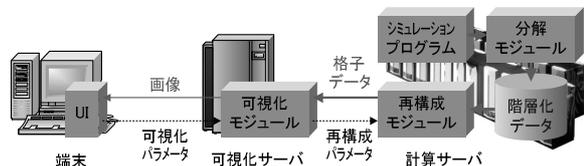


図 6 階層化データ経由の実時間可視化システム

構造のまま扱え、可逆な階層化である。この中でも様々な方式があるが、その中で最も汎用的なのが MT ( Multi-Triangulation ) または MC (Multi-Complex)[8]-[13]という方式で、主に地形データ、ポリゴンオブジェクトに適用されている。

#### 5. 本研究の方式

本研究では、最も汎用性の高い格子データ形式である、非構造四面体格子データの階層化を実現するため、MCを採用した。

MCでは階層構造を作り出すために、局所的な単純化を誤差の小さい順に繰り返し行い、格子が十分単純化されるまで続ける。本研究では図 7に示す頂点除去を単純化操作として用いた。また、誤差は除去頂点の移動距離と定義し、格子上の物理量には

依存しない定義とした。こうすることで、格子と物理量を独立に階層化することができ、例えば、格子が変化しない時系列データの場合、格子は初回のみ階層化すればよくなる。単純化毎に、詳細化データ、誤差、以前の単純化操作との依存関係を記録しておく。これをもとにオリジナル格子データは、図 8 のような詳細化操作の依存関係グラフの形で階層表現できる。

依存関係に基づく順序で、都合に合った詳細化操作を行えば、格子データを様々に再構成できる。例えば図 9 の再構成範囲で詳細度を A まで高めたい場合、a の詳細化操作を依存関係に違わぬ順序で行えばよい。さらに詳細度を B まで高めたい場合は、a と b の詳細化操作を行えばよい。この自由度の高い再構成によって、可視化に重要な部分は詳細度を高く、不要な部分は詳細度を低くした、可視化に効率的な格子データを生成できる。

## 6. 可視化結果

本方式の可視化への有効性を確認するため、図 10 のゴルフボール格子を用いていくつかの評価を行った。図 11 がその結果である。左は約 80 万頂点のオリジナル格子データを可視化したもので、これを 4 万頂点 (元の 5%) まで分解し、階層化データを生成した。真中は階層化データの単純化格子を可視化したものである。これよりボール表面の一部領域を詳細化し、約 900 頂点 (0.1%) 増加させた再構成格子データを生成した。これに見るように、少ない詳細化操作により十分な可視化精度を達成できることがわかる。

## 7. 処理時間について

データ階層化の有効性を述べるためには、分解と再構成、および再構成格子データの可視化に要する処理時間も調べる必要がある。

図 12 は大規模格子データの直接可視化時間と、階層化データの可視化時間の比較を模式的に記したものである。階層化せずに直接可視化した場合、その処理時間は図種にもよるが、だいたい  $O(N^{1/3}) \sim O(N)$  であり、格子データの規模増大に伴い、顕著に長くなる。ここで、 $N$  は格子データの要素数である。一方、階層化データからの再構成時間は  $O(M \log M)$ 、得られた格子データの可視化時間は  $O(M^{1/3}) \sim$

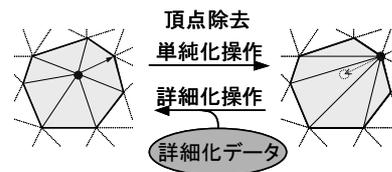


図 7 単純化操作と詳細化操作  
二次元で図示。

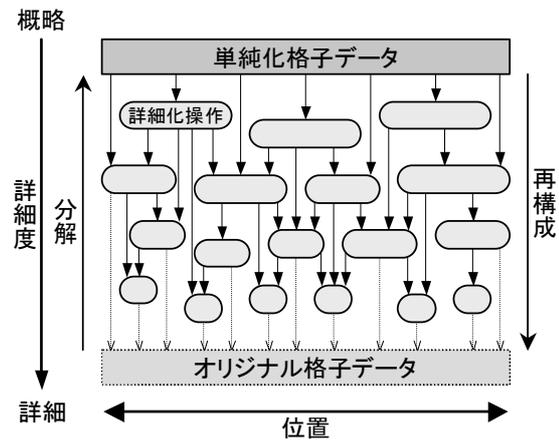


図 8 詳細化操作の依存関係グラフ

ノードの幅は詳細化操作の空間的な影響範囲を表す。空間の位置および範囲は一次元で模式化。

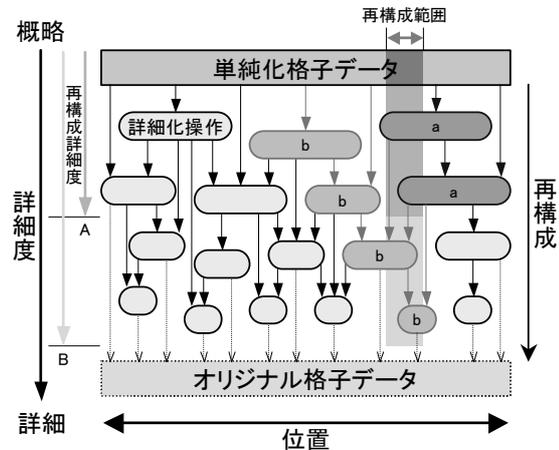


図 9 再構成に関わる詳細化操作

詳細度を A まで高める場合は a の詳細化操作を、詳細度を B まで高める場合は a と b の詳細化操作を行う。

$O(M)$  である。ただし、 $M$  は再構成格子データの要素数である。可視化画像の画素数が一定であることから、可視化に必要な  $M$  は概ね一定である。従って、再構成可視化時間は  $O(\log M)$  となり、規模に対してほとんど変動しない。以上より、格子データが大規模であるほど、データ階層化の有効性が増すことがわかる。

ところで、格子データが小規模である場合は、直接可視化の方が速い。従って、ある規模以上でない

と再構成可視化に利点がないことになる。この規模を最小有効規模と定義する。最小有効規模が小さいほど、データ階層化の有効性が高まる。最小有効規模は、再構成の処理性能以外に、可視化の図種にも依存する。ボリュームレンダリングなど高コストの可視化であれば、最小有効規模は小さくなる。

図 13は分解も含めた可視化時間の比較を模式的に記したものである。分解時間は  $O(N \log N)$  であり、ほぼ線形オーダーであるとはいえ、直接可視化よりも規模依存性があり、若干不利である。しかし、可視化と異なり、分解は初めに一度だけ行えばよい点で有利である。分解による遅れは、複数回の再構成可視化によって取り戻すことができる。合計時間が逆転する回数を最小有効反復数と定義する。最小有効反復数より少ない回数の可視化では、階層化は有効ではなくなる。最小有効反復数は、格子データが大規模であるほど、可視化図種が高コストであるほど、また、分解および再構成の処理性能が高いほど少なくなる。

本研究のプロトタイプにおける最小有効規模、最小有効反復数についての結果詳細は、当日発表にて報告する。

## 8. おわりに

大規模データの効率的な可視化を目指し、三次元非構造格子データの階層化を実現した。概ね良好な可視化画像が得られているが、インタラクティブな可視化のためには再構成の処理速度が十分ではない。今後、階層化データに対するアクセスの効率化、および再構成精度の正確化などにより再構成の高速化を行い、最小有効規模と最小有効反復数の一層の低下を目指す。

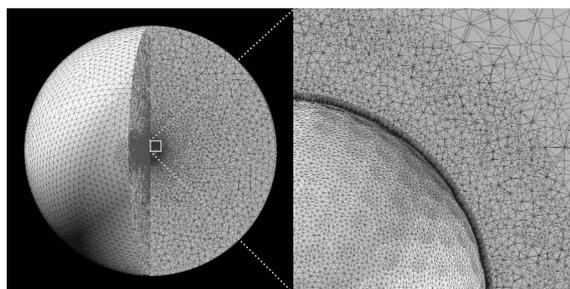


図 10 評価に使用した非構造格子  
頂点数800961, 四面体数4576246.  
(東北大学大学院工学研究科中橋研究室提供)

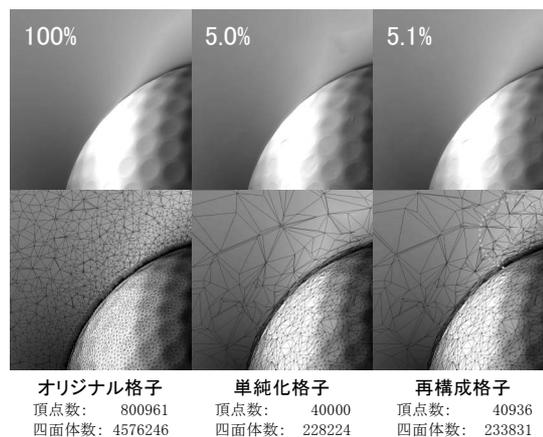


図 11 再構成の例  
白点線で囲まれた部分を詳細化。  
画像は圧力分布を示している。

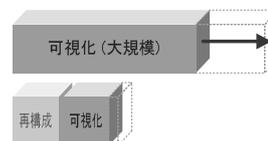


図 12 階層化データの可視化時間  
規模依存性が小さい。



図 13 分解も含めた階層化データの可視化時間  
何度も可視化する場合に有利。

## 参考文献

- [1] P. C. Wong and R. D. Bergeron. "Hierarchical Representation of Very Large Data Sets for Visualization Using Wavelets." *Scientific Visualization: Overviews, Methodologies & Techniques*, 415-428, 1997.
- [2] S. Muraki. "Volume Data and Wavelet Transforms." *IEEE Computer Graphics and Application*, 13(4): 50-56, 1993.
- [3] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery and W. Stuetzle. "Multiresolution Analysis of Arbitrary Meshes." *SIGGRAPH '95*, 173-182, 1995.
- [4] M. Lounsbery, T. DeRose and J. Warren. "Multiresolution Analysis for Surfaces of Arbitrary Topological Type." *ACM Transactions on Graphics*, 16(1): 34-73, 1997.
- [5] L. A. Freitag and R. M. Loy. "Adaptive, Multiresolution Visualization of Large Data Sets using a Distributed Memory Octree." *Proceedings of SC99: High Performance Networking and Computing*, 1999.
- [6] F. Sevilgen and S. Aluru. "A Unifying Data Structure for Hierarchical Methods." *Proceedings of SC99: High Performance Networking and Computing*, 1999.
- [7] A. V. Gelder, V. Verma and J. Wilhelms. "Volume Decimation of Irregular Tetrahedral Grids." *IEEE Computer Graphics International 1999*, 222-230, 1999.
- [8] P. Cignoni, C. Montani, E. Puppo and R. Scopigno. "Multiresolution Representation and Visualization of Volume Data." *IEEE Transactions on Visualization and Computer Graphics*, 3(4): 352-369, 1997.
- [9] E. Puppo and R. Scopigno. "Simplification, LOD and Multiresolution. Principles and Applications." *Eurographics '97 Tutorial Notes*, 1997.
- [10] L. De Floriani, P. Magillo and E. Puppo. "Building and Traversing a Surface at Variable Resolution." *Proceedings IEEE Visualization '97*, 103-110, 1997.
- [11] L. De Floriani, P. Magillo and E. Puppo. "Efficient Implementation of Multi-Triangulations." *Proceedings IEEE Visualization '98*, 1998.
- [12] L. De Floriani, P. Magillo, F. Morando and E. Puppo. "Dynamic View-Dependent Multiresolution on a Client-Server Architecture." *Computer-Aided Design Journal*, 32(13): 805-823, 2000.
- [13] E. Danovaro, L. De Floriani, P. Magillo and E. Puppo. "Representing Vertex-Based Multiresolution Simplicial Complexes, *Lecture Notes in Computer Science, special issue on Digital and Image Geometry*, 2000.