

# 大規模非定常有限要素解析結果のウォークスルー可視化

2007年3月修了  
人間環境学専攻 56812 山室 弥生  
指導教員 吉村 忍 教授

Nowadays the application area of numerical simulation is growing rapidly. Many researchers challenge complicated and large-scale simulation, for example seismic response simulation. The utilization of ADVENTURE system on the Earth Simulator enables such a time dependent simulation of a large-scale model. As this technology advances, post processing requires further advancement. We should think about how to deal with a huge amount of data. In particular, visualization demands interaction with users' needs. One of the solutions of those demands is to develop a Walkthrough System in a precise simulation world. In this research, we propose a Walkthrough System and Parallel Visualization for structural simulations of a large-scale and complex model.

**Key Words:** Visualization, Walkthrough, Finite Element Method, Domain Decomposition Method, Parallel Visualization

## 1. 緒言

近年、数値シミュレーションが対象とする分野は多岐にわたっている。また扱う分野の拡大に加え、地球シミュレータなど高速計算機の発達に伴い、複雑かつ大規模な構造物の地震応答解析などの時系列を伴った解析も可能となってきた。例として、オープンソース並列有限要素法による解析ソフトADVENTURE<sup>1)</sup>による構造物の解析があげられる。現在、有限要素法による解析では精密な解析結果を得るために、数千万から億単位の自由度で行われるようになっている。これらの解析技術の向上に伴い解析結果を表示するための可視化処理<sup>2)</sup>に関してもより大きな発展が必要となってきた。特に、現実の大規模複雑構造物に関しては複雑であるが故にどの部分に問題が潜んでいるかを予測することは難しい。従って、可視化では構造物の中を探し回って問題を見つけ出すデータマイニングの環境が必要になっている。その環境の中では、ユーザの視点移動に合わせ大域的・局所的な表示が同時にできるインタラクティブ性を持ったシステムの構築が重要なポイントである。

これらのニーズを実現する可視化手法のひとつにウォークスルーがある。ウォークスルーではユーザとの対話性が求められるため扱うデータ量に制限がかかってしまう。本研究では大規模構造物解析の可視化手法においてウォークスルーシステムを組み込み、ウォークスルーによってユーザが観察したい位置に移動した後に動解析結果を可視化するというシステムを提案する

ことを目的とし、そのために大規模データの扱い方、並列可視化アルゴリズムについて述べる。

## 2. 大規模解析における可視化

大規模解析の結果を可視化する手法は数多く存在する。それらは大きく3つに分けられる。

クライアント側端末での可視化

計算機サーバ側での可視化

オンライン可視化

表1にこれらの特徴を比較する。本研究では、コモディティPCを利用してのシステム構築、ユーザとのインタラクションを重視しているため、「クライアント側端末での可視化」という視点で研究をすすめる。

Table 1 Comparison of Visualization

Large Scale	( )		
Interactivity		×	
Commodity			×

次に、クライアント側端末での可視化において大規模データを扱うために行われている手法について考える。ここに対する手法には「部分的に可視化」、「データを間引いて可視化」などがあるが、大規模かつ複雑な構造物について対象を丸ごと高精度に解析しているという背景から本研究では、データを間引くことなく対象モデルを丸ごと可視化するという立場で、並列環境を用いたウォークスルー可視化を実現する。

### 3. 大規模データのウォークスルー

本章では、まず、動解析結果を可視化する前にユーザがモデル内を探索し観察地点を選定するまでのある時刻におけるウォークスルー可視化システムについて述べる。可視化のプログラミングについては OpenGL を使用している。

#### 3.1 可視化までの全体の流れ

本研究では ADVENTURE システムを基盤にして行うこととする。全体の流れを図 1 に示す。

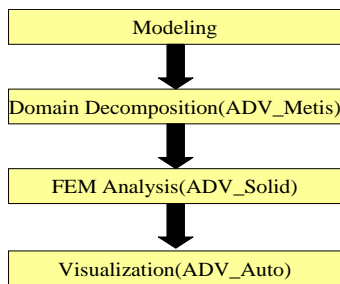


Fig.1 Flowchart of Visualization

大規模構造解析では、並列計算が必要となる。ADVENTURE システムでは、全体モデルデータ作成後に ADVENTURE\_Metis による階層型領域分割が行われ並列計算を行うためのデータ構造に変換される。このデータ構造では Part と呼ばれる部分と、各 Part の下にさらに細かく分割された Subdomain と呼ばれる領域の 2 層構造となっている。領域分割後の有限要素法による並列計算部分では ADVENTURE\_Solid を用いて四面体 2 次要素による解析を行い、変位と応力を出力させた。可視化では領域分割されたモデルデータと解析結果データに対してウォークスルーによる可視化を行う。ユーザは、視点の移動や回転をキーボードによって制御するインタラクティブ機能を持つ。

#### 3.2 データ量

構造物の可視化では、構造物の表面三角形ポリゴンを表示させている。例えば四面体 2 次要素で 2 億自由度の規模の解析では、1000 万枚を超える表面ポリゴンができあがる。1 枚のポリゴンを構成するためには 108 バイトのデータが必要なことから、この規模の表面ポリゴンを扱う場合には約 1GB のデータを扱うことになる。

ウォークスルーを実現させるためには、ユーザとの対話性が重要となり、1 枚の画像を描画する速度は最大でも 0.1 秒である。従って一時的にデータ量を大幅に減らす必要がある。

#### 3.3 ウォークスルーの高速化アルゴリズム

可視化において高速化アルゴリズムは「デー

タ量の削減」、「描画の高速化」がある。本研究が取り入れたアルゴリズムについて詳説する。

##### 3.3.1 可視判定(View Frustum Culling)

可視判定とは、ユーザが見える範囲のものだけを取り出して描画するアルゴリズムである。従って見えない範囲にあるデータをすべて削減できる。大規模データから描画に必要な部分を取り出すには、視点情報からユーザが見える範囲(View Volume)を計算し、検索ルーチンに進む。検索ではオブジェクトを包む Bounding Volume と View Volume の交差判定をし、見えるもののみ抽出する。ユーザの視点の変更されると再度検索プログラムが実行され、ウォークスルーによる可視化が実現される。本研究では Bounding Volume に階層型領域分割法で作成された領域を活用することで、2 段階での可視判定環境を作り、高速化を達成できた。

##### 3.3.2 遮蔽カリング(Occlusion Culling)

可視判定後に抽出された領域に対して、さらに他の領域の影に入ってユーザから見られない領域を削除するアルゴリズムである。これも可視判定同様、Bounding Volume を利用し、Bounding Volume 同士で遮蔽の判定を行う。

##### 3.3.3 LOD(Level Of Detail)

LOD は、ユーザの視点に近い部分は細かいポリゴンで描画し、遠くなるにつれて粗いポリゴンで描画するアルゴリズムである。視点から遠い部分は粗いポリゴンを使うことで少ないポリゴン数で描画するため、データ量が削減できる。

##### 3.3.4 Vertex Array

描画の高速化を図るための手法である。描画をする際に使う OpenGL のライブラリコール数を減らすことによって高速化が可能となる。

##### 3.3.5 Vertex Buffer Object (VBO)

描画の高速化を図るための手法である。このアルゴリズムではグラフィックスカード上にある VRAM を使用する。通常はメインメモリからグラフィックスカードの GPU へ毎回すべての領域データを送信するが、次に描画される領域の一部が前に描画されていた領域と同じ場合、その領域データを VRAM 上に残しておくことで再度データを送られる必要がなくなり、新たに追加される領域分のデータのみ転送で十分となる。したがって描画が高速になる。ただし、この手法は VRAM の容量による大きな制限を受ける。また、毎回全て新しいデータを送り込む必要がある場合にはこの手法は有効ではなく、動解析結果の可視化では使用していない。

## 4. 動解析結果の並列可視化

### 4.1 並列可視化の流れ

大規模な解析モデルの動解析結果を扱う際、データ量の膨大さから1台のPCで可視化を行うことは難しいため、並列計算機環境を利用する。動解析結果の可視化では1 [fps]以上の描画速度が要求される。本研究で並列可視化を行うシステムは8台構成のPCクラスタである。1台をMaster/Slave兼用とし、残り7台をSlaveとして構成している。動解析可視化は、

ウォークスルーで観察視点を固定し、ユーザの視野範囲内にある領域番号を抽出

Masterで領域番号を並列可視化アルゴリズムに従ってSlaveが担当する領域番号に分割  
MasterからSlaveへ視点位置情報と分割された領域番号を送信

各Slaveは受信した視点位置を設定し、担当する領域番号の動解析結果データを1ステップ分読み込み描画

描画した画像をイメージデータとしてMasterに送信

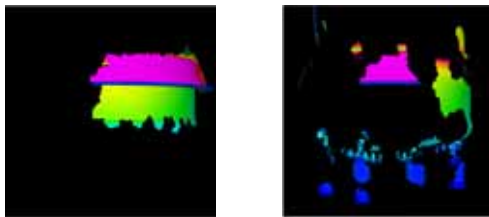
Masterでは各Slaveから送られたイメージデータを合成し1枚の画像として描画

～ 動解析ステップ分繰り返す

という手順で行われる。

### 4.2 並列可視化アルゴリズム

本研究では並列化のためのデータを分割するアルゴリズムとして3つ提案した。



Window partitioning    Model partitioning

Fig.2 Image of parallel visualization

#### 4.2.1 ウィンドウ分割

ピクセル分割とも呼ばれ、結果を表示するウィンドウを構成しているピクセルを8個に分割し、それぞれの画面部分に所属する領域番号に分割する方法である。2つ以上の部分に所属する領域は所属する部分全てにカウントされる。

各Slaveで描画されたイメージは担当したウィンドウ部分のピクセルごとのRGB（色）データのみが送信されるため、データ量は非常に少

ない。あるSlaveが担当した1ステップの可視化結果を図2の左に示す。

#### 4.2.2 モデル分割

領域番号に従って分割する方法である。地球シミュレータ等の大型計算機で行われる大規模解析では階層型領域分割法のデータ構造であるPartの数が1000を超える。そこで、視野内にある領域番号のPartの数字をある規則にしたがって8個に分ける。

この場合、SlaveからMasterに送るイメージデータは全ウィンドウのピクセル分のRGBデータとZ-buffer（深さ）データになる。Z-bufferデータはMasterで画像合成の際に同一ピクセルに2つ以上のRGBデータがあるときにどのデータが手前にあるかを判断するために必要となる。あるSlaveが担当した1ステップの可視化結果を図2の右に示す。

#### 4.2.3 フレーム分割

動解析のステップ番号による分割方法である。各Slaveには最初に抽出された領域番号のデータが全て送られる。その後、Slave0はステップ1、Slave1はステップ2...というように各Slaveで担当するステップをずらし、描画させる。Slaveからは全ウィンドウのピクセル分のRGBデータが送られ、Masterはそのイメージデータをそのまま描画する。ただし、8台のSlaveからMasterへはパイプライン状にデータが送られてくる必要がある。この実装は非常に困難であり、本研究では未実装である。

### 4.3 性能比較・考察

前節のアルゴリズムに対して可視化性能の比較、考察を行う。

[各Slaveが保持すべきデータ量]

並列化なしで9.4GBの場合、各Slaveが持つデータ量は、ウィンドウ分割では9.4 GB、モデル分割では1.2GB、フレーム分割では1.4GBとなる。

[ロードバランス]

並列なしで1台が担当する三角形数を100%とすると、各Slaveが担当する三角形数は、ウィンドウ分割は2.9%～31%まで幅広く値が異なるが、フレーム分割は9.9%～16%となりロードバランスは比較的よいことがわかる。フレーム分割はどのSlaveも同じ枚数を扱うため、ロードバランスは非常によい。

[描画性能]

5章で詳説しているABWRのモデルに対して各並列可視化アルゴリズムのパフォーマンスの計測値を表2に示す。(a：ウィンドウ分割、b：

モデル分割、c：フレーム分割)

Table2 Processing Time (sec)

	Read data from HDD	Transmission	Total
a	0.3 ~ 2.0	0.023	2.37
b	0.55 ~ 1.06	0.427	1.84
c	0.4 ~ 0.5	0.023	0.7

並列なしの場合で動解析結果を描画させたときはHDDからの読み込み時間が大きく影響し、1ステップあたり3～5秒である。それと比較するとウィンドウ分割ではロードバランスが悪いため並列化効率が出ない。モデル分割はHDDからの読み込み時間は短くなったが、通信時間が長くなっている。フレーム分割は十分な性能を持つが、実装が困難である。したがって、欠点はあるがモデル分割とフレーム分割が有効なアルゴリズムであると判断できる。

## 5. 可視化モデル

Table3 Model Data (M=million)

	DOFs	Elements	Polygons
1	35M	7.5M	1.9M
2	203.7M	39.7M	15.4M

### 5.1 モデルデータ

本研究では、表3に詳細を述べる可視化モデルを使用しシステムの性能を確かめた。表3の1はABWRの解析モデル、2はBWRの解析モデルである。

### 5.2 BWRに対するウォークスルーの描画計測

図3に描画計測結果を示す。all\_dataとは何も高速化アルゴリズムを入れていない場合、VCは可視判定、OCは遮蔽カリング、VAはvertex

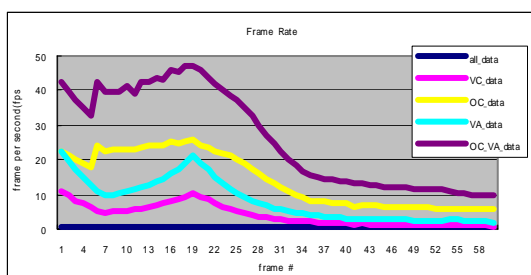


Fig.3 Frame rate of model

arrayを意味する。この結果により、all\_dataの場合0.6fpsだった描画性能が、OC + VAによって10fps以上の描画性能を実現でき、十分ウォークスルー環境を満たしていることがわかる。

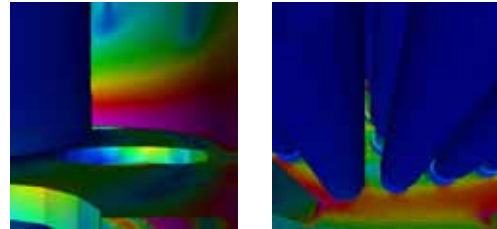


Fig.4 Visualization in BWR

### 5.3 ABWRに対する動解析結果の可視化

本研究の並列可視化システムでは、ABWRのモデルに対して現段階で、0.5～1fpsの描画速度となっている。並列を行わず、1台で動解析結果の可視化を行う場合は0.1～0.3fpsであったため、並列による描画速度の向上は実現された。動解析可視化例を図5に示す。

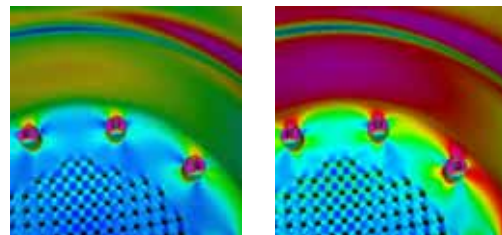


Fig.5 Visualization for Dynamic Simulation

## 6. 結言

本研究では、大規模構造物シミュレーションの解析結果の表示手法としてウォークスルーシステム導入の意義と、システムを構築するための描画の高速化アルゴリズム、並列可視化アルゴリズムを示した。それらアルゴリズムを取り入れたシステムでは3500万、2億自由度の大規模なモデルに対しても十分な描画性能を持つと判断でき、大規模解析におけるポストプロセスに対して新しい手法を提案できた。

## 文献

- 1) <http://adventure.q.t.u-tokyo.ac.jp>
- 2) 河合浩志、吉村忍：ADVENTUREにおけるプレポストツールキットの開発、計算工学会講演会論文集、Vol.9、pp.883-884、2004