

### 3. 2. 2 大規模数値解析結果の先端可視化技術の開発

#### (1) 業務の内容

##### (a) 業務の目的

地震被害評価は大規模数値解析の計算結果に基づくが、これには膨大な量の計算結果を効率的に可視化することが必須となる。3次元視とマルチスケールが可能な可視化を開発するとともに、応急評価に使えるよう高速処理も併せて実現する。

##### (b) 平成25年度業務目的

前年度に開発した都市モデルを利用することで、東京23区全体を対象とした都市地震被害の先端可視化技術を開発する。サブプロジェクト③と連携して、災害対応能力の向上に有効な可視化の方法についての検討を継続する。

##### (c) 担当者

所属機関	役職	氏名	メールアドレス
東京工業大学情報理工学研究科	教授	廣瀬壮一	
東京工業大学建築物理研究センター	教授	坂田弘安	
東京工業大学建築物理研究センター	准教授	山田 哲	
東京工業大学総合理工学研究科	教授	山中浩明	
東京工業大学総合理工学研究科	教授	盛川 仁	

#### (2) 平成25年度の成果

##### (a) 業務の要約

- 1) 前年度に開発した都市モデルを利用することで、東京23区全体を対象とした都市地震被害の先端可視化技術を開発した。
- 2) サブプロジェクト③と連携して、災害対応能力の向上に有効な可視化の方法について検討した。

##### (b) 業務の成果

#### 1) 都市地震被害の先端可視化技術の開発

##### a) 背景

地震発生直後の迅速な構造物群の被害の把握は、効果的な対応策を考える上で重要である。高密度な観測網を都市全体に敷くことは経費がかかるため、限られた観測点で得られる情報を基に被害を推定する方法が必要となる。局所的な地盤特性や個々の構造物の特性が構造物被害を大きく左右するため、被害推定にはこれらの情報を反映することが望ましいとともに、地震直後の初動対応の判断支援をするため、短い時間で結果が得られることが望ましい。

都市を構成する構造物や地盤などの地理情報データや、それらの地震応答を計算する物理ベースの数値シミュレーション手法が開発されている。そこで、地震直後に得られる観

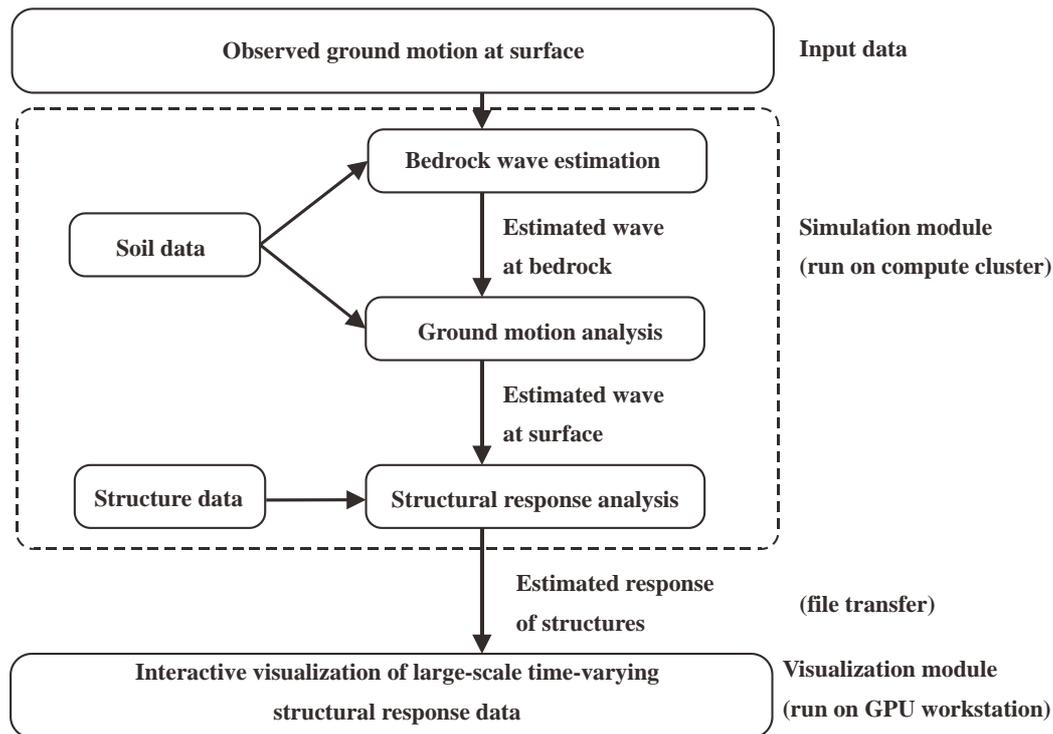


図 1 即時被害推定システムのフロー図

測地震波を入力に、これらの地理情報データと地震応答解析手法を組み合わせた都市の応答シミュレーションを行うことでより合理的な即時被害推定ができる可能性がある（図 1）。また、地震シミュレータ（IES）（Hori and Ichimura, 2008<sup>1)</sup>; Hori, 2011<sup>2)</sup>）を用いた将来の想定地震に対する都市の被害を、個々の建造物の特性まで計算に取り入れて実行することも可能になっている。このように計算機の大きな進展により、即時被害推定や想定地震による都市被害予測を行える環境が整いつつあるが、その計算は膨大になるため、計算結果を効果的に解釈するための可視化技術が必要である。特に、行政などがシミュレーション結果をもとに対策を考案するには、時系列で変化する建造物の変形・損傷を、都市全体の被害分布から各小街区の詳細な被害までシームレスに把握する必要がある。このような結果の把握を短時間で効果的に行うには、大規模な時系列シミュレーションデータのインタラクティブな可視化が必要となる。現状の都市地震シミュレーションでは時間に余裕があるために、GPU 搭載ワークステーションでのバッチ処理による動画作成を通して被害像を把握している。この方法でインタラクティブにデータを可視化しようとした場合、時系列方向に 1 ステップ進めるたびに静的データの可視化に必要な時間と同等の時間がかかってしまう。注目の時間を巻き戻して再生し、また、視点を変えて再度再生するなどの操作が意思決定に重要にもかかわらず、動的データのインタラクティブ可視化が即時被害把握のボトルネックとなっている。一方で、time critical rendering のためにデータの簡易化方法（Level Of Detail を調整）が開発されており（Gobbetti and Bouvier, 1999<sup>3)</sup>; Li and Shen, 2002<sup>4)</sup>）、これと、現状の GPU 搭載ワークステーションのハードウェア特性（ファイルシステム読み込み速度、主メモリー-GPU メモリ間転送速度）に合わせた手法を組み合わせることで、都市震災シミュレーションの可視化性能を向上できる可能性がある。

以上を踏まえ、本研究では、即時被害推定システムのデータの簡易化とオフラインレンダリングのハードウェア特性に適した可視化手法により構造物の時系列応答の実用的なインタラクティブ可視化手法を開発する。

## b) 対象とする可視化データ

対象とする可視化データは東京23区の建物と地形に関するデータで、詳細は以下のとおりである。

- ・建物データ：三角形要素で構成され、各節点上に3次元float型の変位ベクトルが設定される。時系列で値が変わるのは変位ベクトルのみで、三角形要素を構成する節点数、要素数、接続情報は時系列で変わらない。GISタイル単位で作成されており、全データを読み込むと23区全体を示す建物データとなる。東京23区全体の建物データの諸量は以下のとおりである。

節点数： 2,700万点 (26,896,326)

要素数： 5,200万個 (52,428,306 三角形)

物理量： 変位データ (float型3次元ベクトルデータ)

時刻ステップ： 数万ステップ (10,000~30,000)

- ・地形データ：三角形要素で構成され、各節点上にfloat型の標高データ (スカラー) が設定される。本データは時系列では変わらない。東京23区全体の諸量は以下のとおりである。

節点数： 3,400万点 (34,399,785)

要素数： 3,400万個 (34,200,000 三角形)

物理量： 標高データ (float型スカラーデータ)

本研究では、以上の建物データと地形データを、時間ステップを進めながらリアルタイムでアニメーション表示することを目的とする。

## c) 都市地震被害の先端可視化技術の概要

従来の簡便な被害推定手法に対し、構造物毎に時系列計算を行う被害推定は大規模データの可視化が必要となる。既往の大規模可視化手法はスパコンなどシミュレーション計算を行う計算機を使ったオンラインレンダリング (Ahrens *et al.*, 2009<sup>5)</sup>; Tu *et al.*, 2006<sup>6)</sup>) と、GPU 搭載ワークステーションなど、シミュレーション計算を行う計算機とは別の計算機を使ったオフラインレンダリングに大別できる (Meyer and Wischgoll, 2006<sup>7)</sup>)。オンラインレンダリングによるバッチ処理・インタラクティブ可視化は高性能だが、スパコンを占有する必要性があり、特にインタラクティブ可視化はシステムが複雑になるなど、限られたユーザにしか使えないのが現状である。一方で、オフラインレンダリングは、性能はある程度であるものの、簡便・低価格で占有利用できるもので、即時被害推定に向いている。幸い、即時被害推定で必要となる可視化データの量は多くてもテラバイトオーダーなので、オフラインレンダリング手法の性能を向上させることで対応できる可能性がある。

そこで、本研究では従来のオフラインレンダリング手法を動的データのインタラクティブ可視化に使った際の問題点を整理した上で、データの簡易化方法と、オフラインレンダリングのハードウェア特性 (ファイルシステム読み込み速度、主 CPU メモリ-GPU メモ

リ間転送速度) に合わせた手法を組み合わせることで、従来の広域震災シミュレーションの可視化性能の向上を図る。

本研究で用いた計算機の作業環境は以下のとおりである。

CPU : Intel(R) Xeon(R) CPU E5-2680 2.7 GHz (8 コア)×2

OS : RedHat Enterprise Linux 6.2 (x86\_64)

メモリ : 128Gbyte

グラフィックス : Nvidia Quadro 6000

ソフトウェア : AVS/Express Version 8.1 (Linux\_64\_el6)

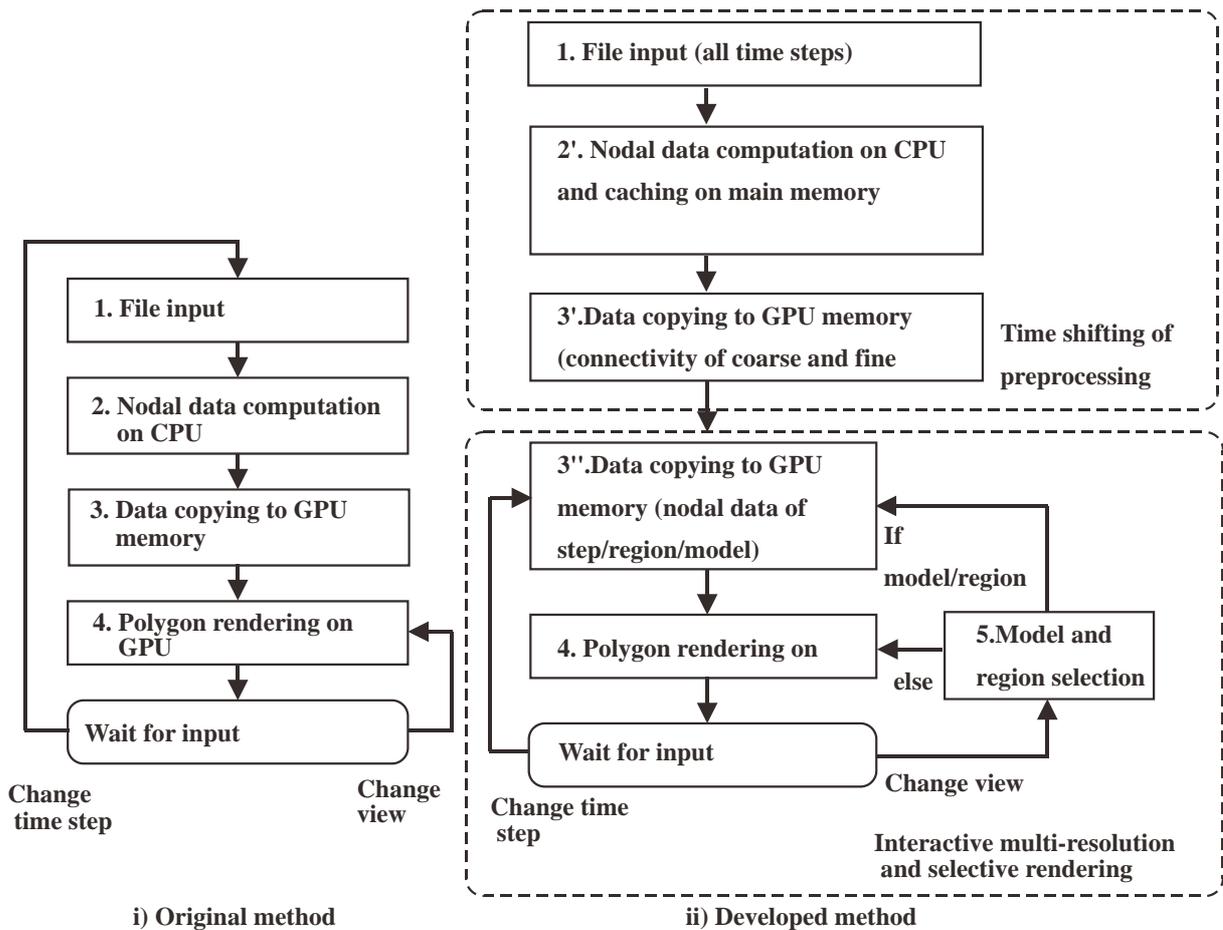


図 2 可視化手法の比較

通常のオフラインレンダリングでは、1. ファイルシステムからのデータ読み込み、2. CPU による節点値 (座標・色・法線方向) の計算、3. GPU への節点値の転送、4. GPU 上でのポリゴンのレンダリングを繰り返すことで可視化を行うことが一般的である (図 2 i)。これらの 4 つのプロセスの内、4. のレンダリングによる視点変更にかかるプロセスにはあまり時間がかからない。よって、一タイムステップの静的状態で視点を変更することは 4. のみを再計算することによってインタラクティブに実施することが可能である。一方で、タイムステップを進めるには 1.~4. をステップ数分繰り返すことになり、多くの計算時間が必要となり、動的データのインタラクティブ可視化は困難になる。理想的には、全ステップの節点情報を GPU メモリ上に置いておき、該当ステップのみレンダリングすること

で動的データを 4. のフレームレイトで可視化できるはずである。しかし、ステップあたりの節点情報はノード数\*10\*sizeof(float) bytes であり、全てのステップの情報をあらかじめ現状の GPU メモリ上に載せておくことはできない。そこで、図 2 ii)に示すように、インタラクティブ可視化作業中に必要なステップの分だけメインメモリから GPU メモリに転送・レンダリングを行うことで、動的可視化のフレームレイトを向上させる。ここでは、メインメモリから GPU メモリに節点情報を転送する時間はかかるものの、インタラクティブ可視化前に 1. と 2.を全ステップ分行い、節点情報をメインメモリ上に置いておくことで、通常の可視化方法に比べて大幅にフレームレイトを向上させることができる。また、メインメモリ上に節点データが置いてあるため、巻き戻し再生時など、節点情報を繰り返し使うことができ、その都度ファイルアクセス、節点情報を計算する方法に対し大幅な高速化が可能となる。

以上の time shifting of preprocessing によるデータ転送時の工夫を行う際、3.の主 CPU メモリ-GPU メモリ間での転送が可視化作業の律速となる。3.の時間は節点数に比例しているため、節点数を削減することがさらなるフレームレイトの向上に対して効果的である。そこで本研究では、ポリゴンから十分離れた視点からは形状を単純化して見せる簡易モデルを使った多重解像度レンダリング (multi-resolution rendering) と、画面外に映らないポリゴンをあらかじめ判定して GPU に転送しない選択的レンダリング (selective rendering) を使って GPU に転送するデータ量を削減する。これらの工夫のほか、本研究では、操作端末と表示端末の分離、視点情報入力による自動位置合わせ、地名情報表示などの機能を開発して、可視化の高速化と利便性の向上を図った。下線で示したそれぞれの要素技術についての詳細を以下に示す。

#### d) 開発した要素技術

##### i) 多重解像度レンダリング (multi-resolution rendering)

地理情報システム (Geographical Information System、GIS) に蓄積された建物の形状データから自動構築されたモデルでは多くの節点と要素が生成される。そこで、主 CPU メモリ-GPU メモリ間での転送時間を減らすためには節点数の削減が重要となる。ここでは適切なデータの縮減を行い、視点により簡易モデルと詳細モデルを切り替える、多重解像度レンダリング (multi-resolution rendering) を実装した。

詳細モデルでは、各構造物の天井を複数頂点からなるポリゴンを三角形分割し、壁面をそれぞれ階毎に三角形としてモデル化する。各階毎の変位はタイムステップ毎に同じ値となる。簡易モデルでは、図 3 に示すように、天井を四角形として近似し、階数を間引いてモデル化する。

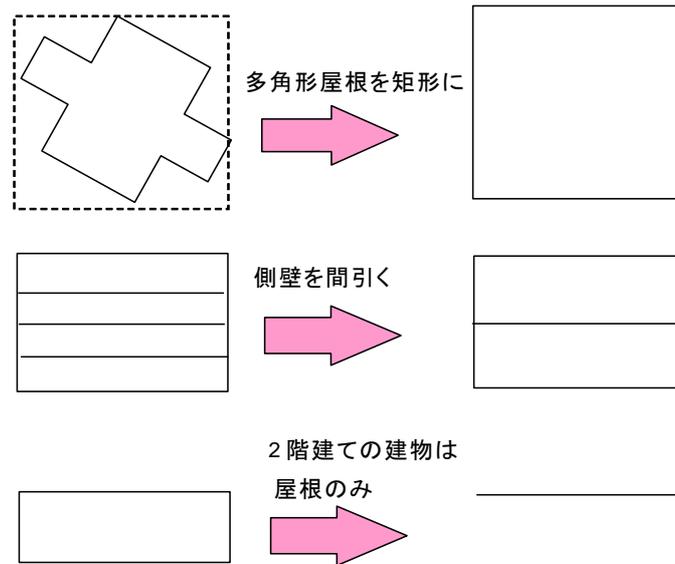


図 3 簡易モデルの作成

このような簡易モデルの作成により、表 1 に示すように、東京 23 区全体のデータは節点数で約 34%、要素数は 21%まで削減できることがわかった。なお、モデルの切り替えは、全体表示時には簡易モデル、区単位の場合は詳細モデルを用いて表示するように実装した。

表 1 詳細モデルと簡易モデルの比較

	詳細モデル	簡易モデル	データ削減率(%)
節点数	30,879,002	10,568,852	34.2
要素数	46,874,319	10,052,968	21.4

## ii) 選択的レンダリング (selective rendering)

可視化の対象とする東京 23 区の建物と地形に関するデータは予めタイル状に分割された領域ごとに並べられている。したがって、タイル状の領域単位でデータをメモリに保持し、図 4 に示すように、回転や拡大操作時において可視化の視野画面内に含まれるタイル状の領域に関するデータのみを転送・レンダリングする。これによって可視化の処理を高速化できる。あるタイル状の領域が視野画面に含まれているか否かの判定は視点変更時毎に必要となるが、簡単な算術で判定することができるため、判定時間はレンダリング時間に比べて無視できる。

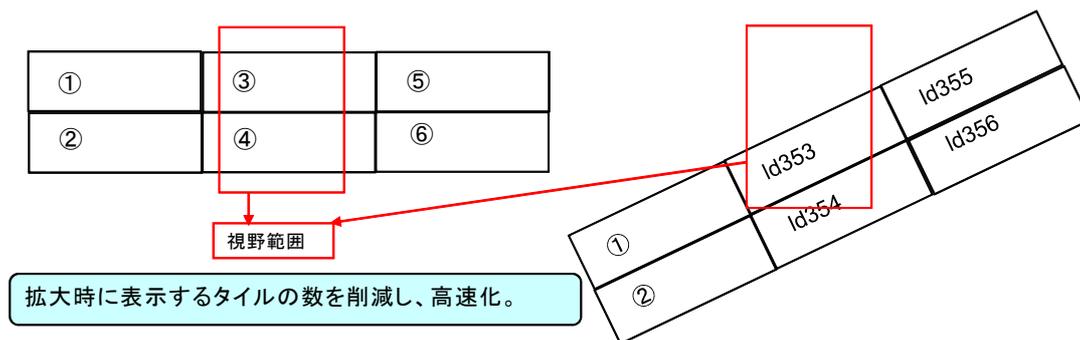


図 4 タイル状の領域と視野画面の関係

### iii) データ転送時の工夫 (time shifting of preprocessing)

可視化には OpenGL を用いるが、GPU 上における Vertex Buffer Object (VBO) と呼ばれるバッファを用いた OpenGL の呼び出し方法を採用することによって描画の高速化を図った。

まず、図 2 ii) に示すように、インタラクティブ可視化前にまとめて行うことができる 1. ファイル読み込みと 2. 節点値の CPU 計算をあらかじめ行っておく。ここでは、あらかじめ全ステップ分データ読み込み、節点値を計算した上で、メインメモリ上にキャッシュする。また、多重解像度レンダリングのための簡易モデルの節点情報もこの段階で計算し、メモリ上にキャッシュしておく。

次に、時系列で変化のない節点や要素の接続情報を初回のみ GPU に転送する。時系列再生時には、変化情報 (頂点位置座標 XYZ、頂点色 RGB、法線ベクトル) のみを GPU に転送することで、転送するデータを削減する。ただし、回転などの幾何変換操作時にはデータの転送を行わない。

表 2 は OpenGL の呼び出し方法を VBO に変更した改良前後による処理速度の結果を示したものである。東京 23 区全体のデータを用いた場合、描画速度 fps (frame per second) で約 10 倍の高速化が図られており、回転などの幾何変換操作時も、東京 23 区全体の場合に 35fps となることを確認している。

表 2 VBO を用いた改良前後における描画速度

データ	節点数	要素数	改良前の fps	改良後の fps
東京 23 区全体	30,879,002	46,874,319	0.1	1.0
新宿区	2,14,789	3,477,160	3.0	11.5

### iv) 操作端末と表示端末の分離

操作端末と表示端末を分離して、街区ごとに端末を分離して表示できるようにした。図 5(a)~(d) に示すように、操作端末には Windows 計算機を用い、表示端末には Linux 計算機を用いた。これにより、操作性を改良するとともに描画速度を向上させた。なお、今回は Windows 計算機 1 台 - Linux 計算機 1 台を用いた 1 対 1 のシステムを開発したが、1 台 - 複数台への構成システムへの変更は容易であり、複数の表示端末で複数の街区を描画させることも可能である。

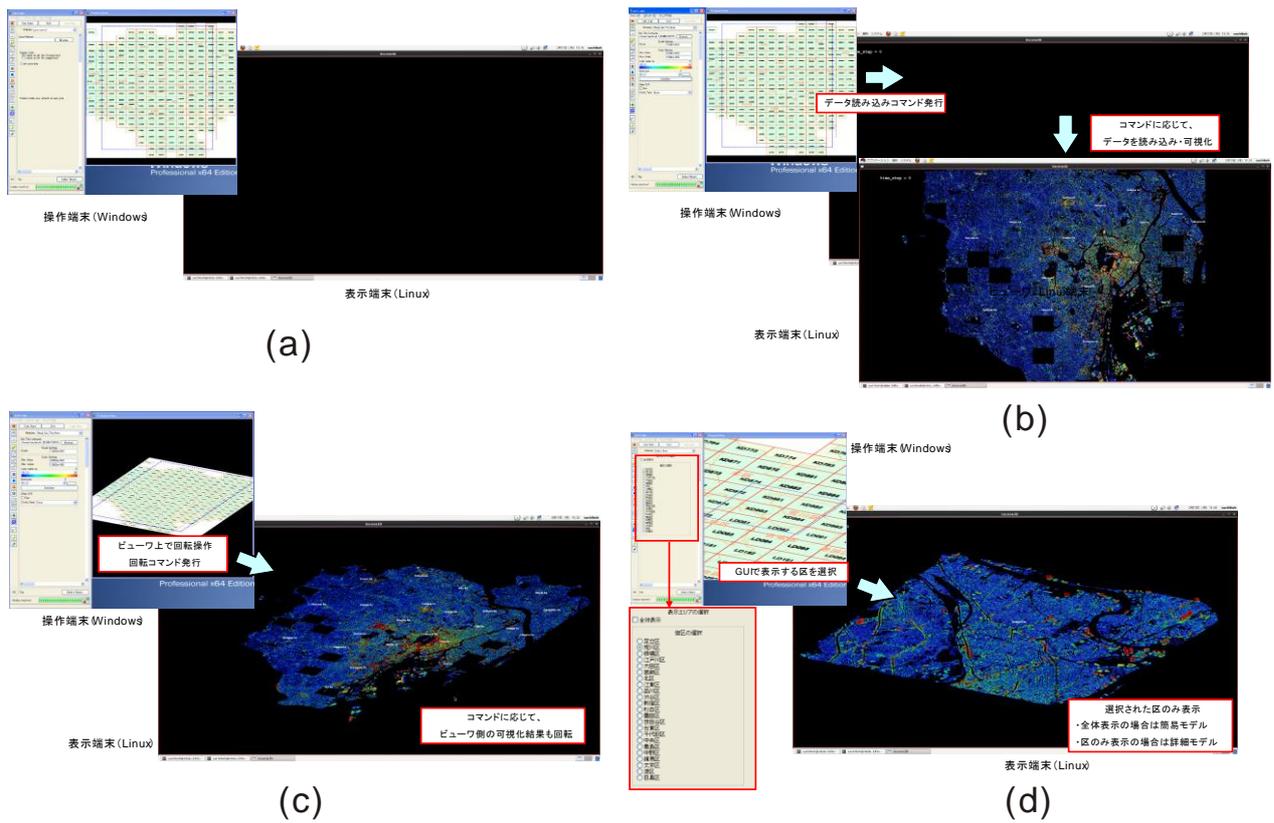


図 5 操作端末と表示端末の分離

#### v) 視点情報入力による自動位置合わせ

ユーザーインターフェースから視点位置と表示範囲を指定できるような機能を有している (図 6(a))。

さらに、緯度・経度で指定された位置に、自動的に視点を合わせる機能を開発した。緯度・経度は以下に示す書式で、視点情報ファイル (\*.view) として与えられるものとする。

35.689444444444 ; 1 行目に倍精度浮動小数点型で緯度を記述

139.691666666667 ; 2 行目に倍精度浮動小数点型で経度を記述

ファイルが指定されたタイミングで、緯度・経度を読み込み、解析データ座標系に変換後、指定位置に視点を合わせる (図 6(b))。

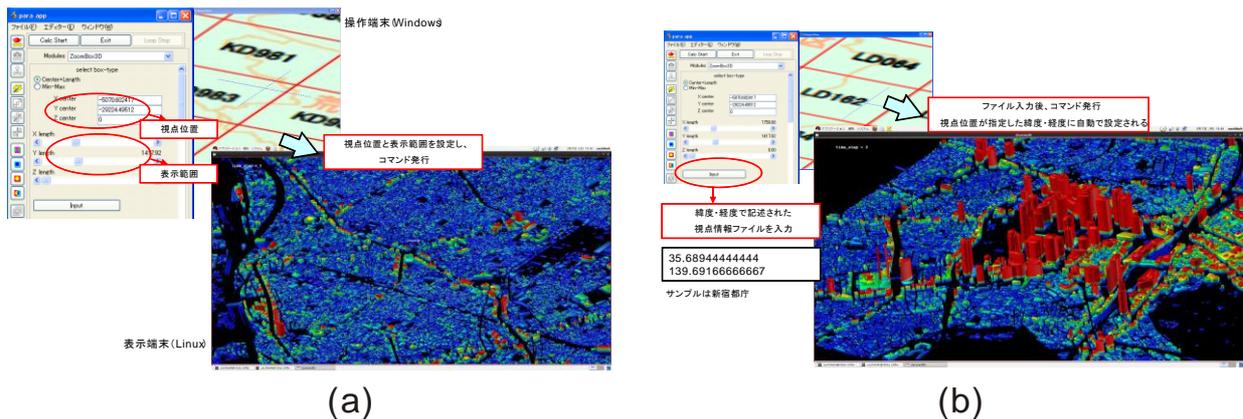


図 6 視点情報入力による自動位置合わせ。(a)ユーザーインターフェースと (b)視点情報ファイルからの入力

#### vi) 地名情報表示などの機能

指定位置に文字列ラベルを表示する機能を開発した。表示位置を示す XYZ 座標をファイル指定、表示するラベルを定義ファイル内に記入することにより、文字列ラベルを 3 次元空間上に表示することができる。なお、表示できる文字列は英数字に限る。

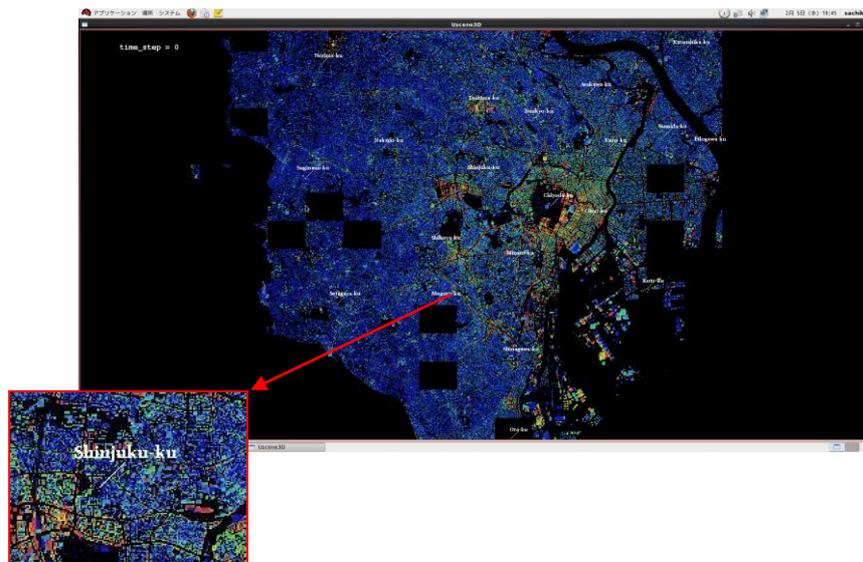


図 7 地名情報表示機能

さらに、解析データに合わせた平面に、画像データを張り合わせて表示する機能も開発した。地図画像など、地名がわかる画像を重ねることで、街区境界などをわかりやすく表示することができる。ただし、画像の位置合わせや座標変換には対応していないので、予め解析データと位置合わせされた画像を準備する必要がある。

#### vii) 新規に開発した可視化モジュール

上述の i)~vi)の機能を実現するために、本研究では以下の AVS/Express 用の 3 つの可視化モジュールを新たに開発した。

- **Read\_Gis\_Tile** モジュール：GIS タイルデータの読み込みを行うと同時に、変位データの RGB 変換、法線生成、三角形ポリゴン情報の生成を行う。さらに、作成した RGB 値、法線、三角形ポリゴン情報を用いて、VBO による三角形の描画処理を行う。
- **Display\_Map** モジュール：指定した平面内に画像データを貼り合わせて表示する。また、指定位置に 3 次元文字列ラベルを表示する。
- **ZoomBox3D** モジュール：3 次元ボックス形状（表示範囲とする矩形領域の XYZ 座標）、または、中心位置と表示範囲（視点中心となる XYZ 座標 + 表示範囲（長さ））の指定により、視点情報の位置合わせを行う。また、緯度・経度情報が記述された視点情報ファイルを入力することにより、視点位置を指定位置に合わせる。

#### e) 応用例

図8に即時被害推定システムの操作例を示す。まず、全体図を簡易モデルで表示した状態で画面が起動する。次に、タイムステップ方向に再生すると、図8 i) ~ i”)のように時刻歴で可視化が進む。ここでのフレームレイトは約30fpsである。次に、マウス操作によりをすると、図8 ii)のようにして視点が変更され、簡易モデルのしきい値を超えると自動的に図8 iii)のように\_詳細モデルに切り替わる。さらにマウス操作でズームインし（図8 iv)、v))、再度タイムステップ方向に再生すると、図8 v)~ v”)のように時刻歴で可視化が進む。ここでのフレームレイトは約60fpsである。このように、本可視化システムは実用的に動的可視化を行うことができる。

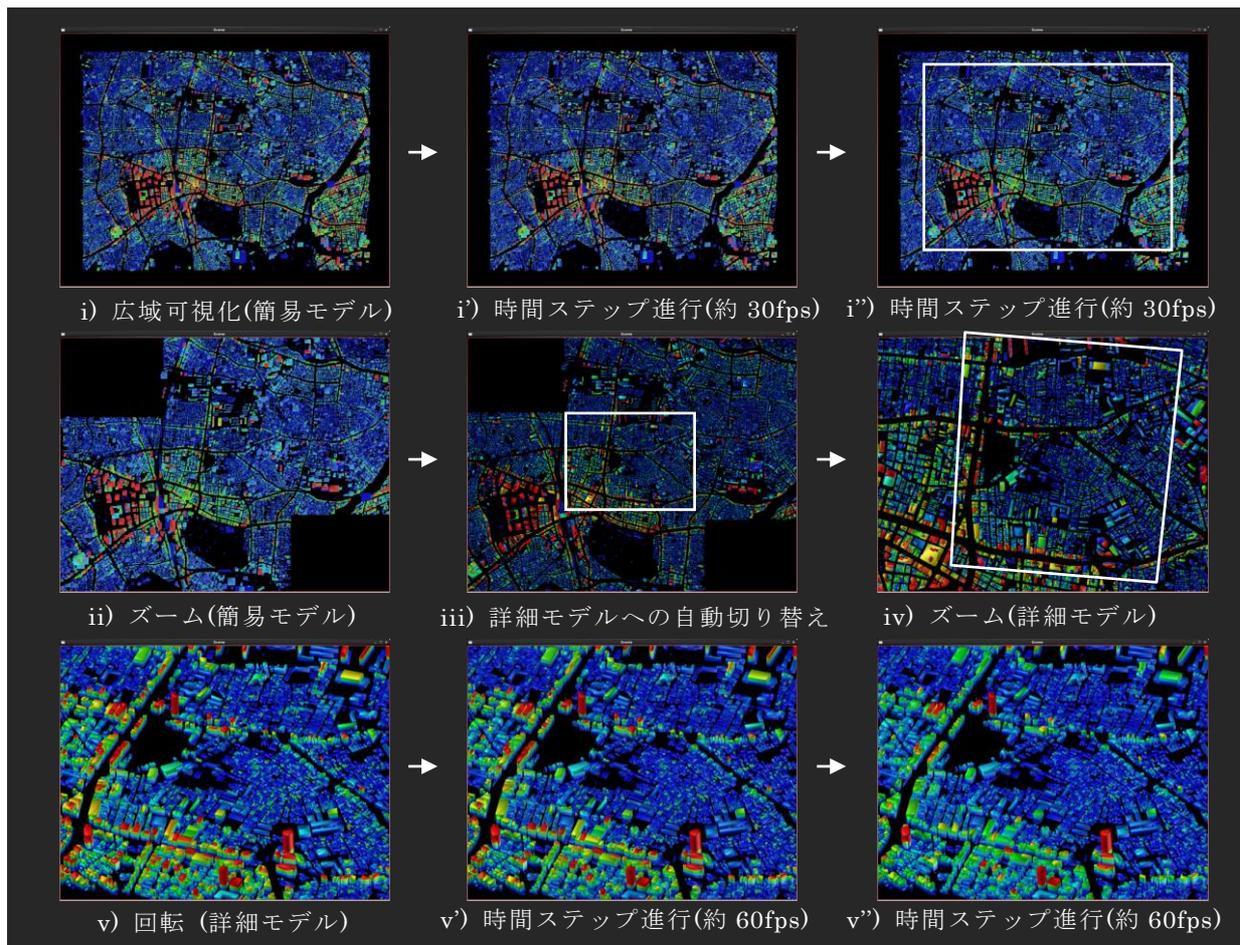


図 8 可視化の応用例

## 2) サブプロジェクト③と連携した可視化の方法の検討

サブプロジェクト③と連携して検討を行った結果、以下の2点が重要であるとの認識で合意した。

- ・地方公共団体などユーザの意見を取り入れるシステムとすること。
- ・火災シミュレーションなどの、地震被害以外の可視化にも対応するシステムを構築すること。

以上の2点については平成26年度以降の研究に反映させる。

### (c) 結論ならびに今後の課題

前年度に開発した都市モデルを利用することで、東京23区全体を対象とした都市地震被害の先端可視化技術を開発した。数値解析データの簡易化とオフラインレンダリングのハードウェア特性に適した可視化手法などにより構造物の時系列応答の実用的なインタラクティブ可視化手法を開発した。また、サブプロジェクト③と連携して、災害対応能力の向上に有効な可視化の方法について検討した。

今後の課題としては、表示スケールに応じた表示情報の切り替えや地図情報との関連付けなど、より高度な可視化技術の開発、及び、操作メニューを極力減らすなどの可視化における操作性の向上が考えられる。ユーザの意見を取り入れつつ、可視化技術の改善を図

る予定である。

(d) 引用文献

- 1) Hori, M. and Ichimura, T.: Current state of integrated earthquake simulation for earthquake hazard and disaster, *J. Seism.*, 12(2), pp.307–321, 2008.
- 2) Hori, M.: *Introduction to Computational Earthquake Engineering*, Imperial College Press, 2011.
- 3) Gobbetti, E. and Bouvier, E.: Time-critical multiresolution scene rendering, In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years, VIS '99*, pages 123–130, Los Alamitos, CA, USA, IEEE Computer Society Press, 1999.
- 4) Li, X. and Shen, H. W.: Time-critical multiresolution volume rendering using 3d texture mapping hardware, In *Proceedings of the 2002 IEEE Symposium on Volume Visualization and Graphics, VVS '02*, pp. 29–36, Piscataway, NJ, USA, IEEE Press, 2002.
- 5) Ahrens, J. P., Woodring, J., DeMarle, D. E., Patchett, J. and Maltrud, M.: Interactive remote large-scale data visualization via prioritized multi-resolution streaming, In *Proceedings of the 2009 Workshop on Ultrascale Visualization, UltraVis '09*, pp. 1–10, New York, NY, USA, ACM, 2009.
- 6) Tu, T., Yu, H., Ramirez-Guzman, R., Bielak, J., Ghattas, O., Ma, K. L. and O'Hallaron, D. R.: From mesh generation to scientific visualization: an end-to-end approach to parallel supercomputing, In *Proceedings of the 2006 ACM/IEEE conference on Super-computing, SC '06*, New York, NY, USA, ACM, 2006.
- 7) Meyer, J. and Wischgoll, T.: Earthquake visualization using large-scale ground motion and structural response simulations, In *Georges-Pierre Bonneau, Thomas Ertl and Gregory M. Nielson, editors, Scientific Visualization: The Visual Extraction of Knowledge from Data, Mathematics and Visualization*, pp.409–432, Springer Berlin Heidelberg, 2006.

(e) 学会等発表実績

学会等における口頭・ポスター発表

発表成果（発表題目、口頭・ポスター発表の別）	発表者氏名	発表場所（学会等名）	発表時期	国際・国内の別
広域震災シミュレーションの可視化に関する基礎検討 （口頭）	藤田 航平，廣瀬 壮一，市村 強，堀 宗朗，盛川仁，山中浩明，坂田 弘安，山田哲	日本地震工学会大会 （渋谷区）	2013年	国内

学会誌・雑誌等における論文掲載

なし

マスコミ等における報道・掲載  
なし

(f) 特許出願, ソフトウェア開発, 仕様・標準等の策定

1) 特許出願

なし

2) ソフトウェア開発

名称	機能
可視化モジュール	データの読み込み、変換、ポリゴン生成や表示を行うための AVS/Express 用のモジュール ((2)(b)1)c)vii)参照)

3) 仕様・標準等の策定

なし

**(3) 平成 26 年度業務計画案**

平成 25 年度までに開発された都市地震被害の先端可視化技術の実用性の検討を行う。社会基盤構造物への拡張など、構造物モデルの高度化も進める。サブプロジェクト③と連携して、火災を含めた、災害対応能力の向上に有効な可視化の方法についての検討を継続する。