

# シミュレーションデータの二つの可視化手法

神戸大学システム情報学研究科 陰山 聡 (KAGEYAMA Akira)

**概要** スーパーコンピュータの演算能力の指数関数的な進歩により、データの可視化に関する技術にも革新が要求されている。ここでは、可視化に関して我々が行っている二つの試みを紹介する。一つは没入型バーチャルリアリティ装置を用いた3次元のかつ対話的な可視化法であり、もう一つは計算を行いながら可視化を行う In-Situ 可視化を発展させた方法である。この方法により、計算が終わった後でも、視点や可視化手法を自由に切り替えることが可能となる。

## 1 はじめに

この論文は、本研究会<sup>1</sup>とその少し後に行った別の講演<sup>2</sup>の内容から、可視化研究に関する部分のみ抜き出してまとめたものです。研究会では、私がこれまで行ってきた地磁気の計算機シミュレーションについてもお話しましたが、それに関しては同じく応用力学研究所の研究会で以前お話した時の講究録 [1] がありますので、ここでは可視化についてのみまとめることにしました。

まずはじめに、そもそも可視化とは何かということについて述べたいと思います。可視化というと、一般的には情報の可視化 (Information Visualization) も含まれますが、ここでの可視化とはいわゆる科学データの可視化 (Scientific Visualization)、あるいは単にデータ可視化を意味することとします。

データ可視化の典型的な例は天気図でしょう。気象通報という番組をご存じでしょうか。NHKのラジオ第二放送で放送している番組で、日本各地のその日の天気をアナウンサーが読み上げる番組です。「石垣島、東北東の風、風力3、晴れ、20ヘクトパスカル、21度」といった気象のデータを延々と読み上げるわけですが、この番組をただ聞いても面白いわけではありません。この番組を聴く人は、アナウンサーが読み上げる各地の気象のデータを専用の地図に書き込んでいきます。アナウンサーの読み上げが終わった後に、地図の上でたとえば気圧の値が同じ値になりそうなところを目で補間しながら結びます。気圧の等高線です。この等高線図をみると低気圧がある場所、高気圧がある場所などがわかります。我々が日頃なじみのある天気図ができあがるわけです。これでその日の大気の状態がわかり、次の日の天気も予測できるようになります。でも考えてみれば、この天気図に含まれている情報は結局のところ、アナウンサーが読み上げた数値のデータに含まれてはいたはずですが、その数値データを見たり聞いたりしただけではわからなかった情報を天気図にするという可視化の作業を加えることで、引き出すことができたわけです。データ可視化の意義と目的はここにあります。数値データに潜む情報を引き出し、それを分かるようになること、それが可視化の一つの目的です (図1)。

<sup>1</sup>九州大学応用力学研究所 共同利用研究集会 非線形波動研究の最前線—構造と現象の多様性— (2012年11月)

<sup>2</sup>第2回神戸大学統合研究拠点サロン (2012年12月) における講演「防災減災と3次元可視化技術」

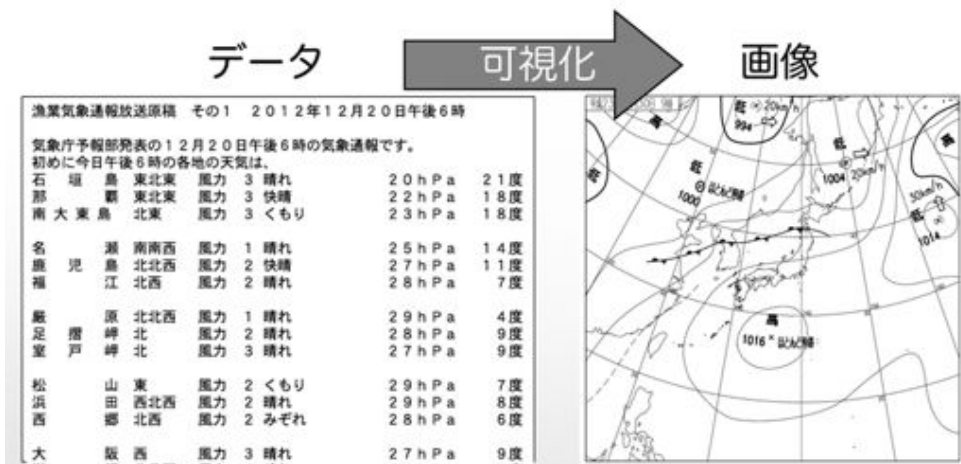


図 1: 可視化の概念。数値データの羅列から画像への変換。

可視化の目的の一つは要するに「分かる」ことだと言えます。これまで理解できなかったこと、気づかなかった現象や構造を「分かる」ようにするのが可視化の役割です。英語で「分かった」というのを“I see.”つまり、「見えた」と言います。何かが見えるようになったとき、我々は何か分かったことになるだと言えらると思います。たとえば計算機シミュレーションで何かの現象を研究している研究者が、計算機の出力データ（それは数値の羅列です）に対して可視化という処理を施して何らかの画像にし、その画像をみて「ああ、わかった」といえたとき、可視化は本質的に重要な役割を果たしています。

可視化の意義にはもう一つあります。それは「伝える」ことです。例えば上記のシミュレーション研究者がある現象について、「この現象は、こういうことだ。」というふうに「分かった」とき、あるいはその人にとって「見えた」とき、それを他の研究者、あるいは一般の人たちにその理解を伝えるという役割です。研究者の頭の中にあるその現象、その理解の仕方を、言葉だけを使って他人に伝えるのは難しい場合が多いものです。精緻な概念を共有する同じ専門家同士であれば専門用語を通じてその理解を伝えることは可能ですが、専門外の人たちや、一般の人たちには通用しません。そのような場合にもやはり可視化が重要な役割を果たします。このような可視化は「伝えるための可視化」だと言えます。

「なぜ」可視化をするのか、という理由として二つ挙げました。次に「いつ」可視化するのか、という問題について考えてみます。

大規模な計算機シミュレーションの場合、「いつ」可視化するかと言えば、それは、計算が終わった後に処理する（ポストプロセス）というのが通常であり、それ以外はあまり考えられなかったのですが、最近のスーパーコンピュータの進歩、特にその演算速度の向上がそのような状況を変えてきました。計算の終わった後で可視化するということが現実には難しくなってきたのです。なぜスーパーコンピュータの演算速度が向上するとポストプロセスとしての可視化が難しくなるのか、そして、その問題を解決するためのどのような方法があるのか、また、我々自身がどのよ

うな方法をとろうとしているのかについては第4章で述べます。

「いつ」可視化するかという問題の次に、「どのように」可視化するかという点について考えてみます。データ可視化というのは一つの研究分野を構成していて、いかにデータを可視化するかについては様々な研究が行われ、様々な可視化アルゴリズムやツールが開発されてきました。無料・有料の様々な可視化ソフトも手に入ります。そのような可視化ソフトは、機能が豊富で、多くの可視化・解析には十分な機能が実装されています。PCで実行することが多いのですが、一部のソフトは大規模な並列計算機でも可視化処理が可能です。

通常の可視化ソフトではできない種類の可視化もあります。特に欠けているのは3次元性と対話性です。PCのモニターのウィンドウを見て、そのマウスを使って解析するのが通常の可視化ソフトを使った可視化なのですが、そのようなスタイルには3次元性と対話性が欠けているのです。3次元性と対話性とはどういったものかを説明するには、その二つを補うために私たちが行っている可視化について説明するのが早いと思います。それが次の章で説明するバーチャルリアリティ技術を用いた可視化手法です。

## 2 バーチャルリアリティ技術を用いた可視化

バーチャルリアリティ (VR) というのはよく映画などにも登場し、一般的にもよく使われる言葉ですので、詳しい説明は必要ないでしょう。簡単に言えば、計算機で構成した人工的な世界をあたかも実在する世界化のように見る人に思わせる技術です。シミュレーションデータを可視化する時には、いわゆるコンピュータグラフィックスの技術を使って3次元世界を構成し、その人工的な世界を仮想的なカメラで映した映像を画像として取り出します。ですからVRとデータ可視化は本来親和性の高いものです。

VRの歴史は古く、様々なルーツがありますが、パイオニアワークとして有名なものはI.E. Sutherlandの“A head-mounted three dimensional display”というタイトルの1968年の論文[2]です。HMDと略称される頭に装着するタイプのヘルメット型、あるいは眼鏡型のディスプレイを初めて実現したものがこの研究です。当時は大がかりな装置で、天井につながれた棒の先に双眼鏡のようにのぞき込むディスプレイがついているものでした。

HMDは現在でもVRを実現するための装置として最もよく使われる画像表示方式です。究極的なVRとは本来、視覚だけに限らず、人間の五感すべてをうまく「だます」ことで、現実にはない人工世界を実在するかのように思わせるものであるでしょうが、現在のところ五感すべてを使ったVR装置というのは実在しません。五感の中でも視覚を使ったVR装置に重点が置かれています。人間が外界を認識する上で、最も重要な役割を果たしているのが視覚ですから、まずは視覚系のVR技術が発達しているのは当然のことでしょう。

HMDは現在では比較的安価な市販製品などもあり、以前よりもずっと身近な存在になりました。しかし、VRの基本装置としてみるとHMDには大きな弱点があります。それは視野の狭さです。人間の視野は角度にしてかなり広く、水平方向には約180度、垂直方向には約130度あります。VR装置として視覚を「だまして」人工世界を提示するには、この視野角すべてをカバーするディスプレイを使うことが望ましいわけです。ところが、現在の標準的なHMDの視野角は水平方

向には40度、垂直方向には30度程度しかありません。最近ではこれよりもずっと広い視野角を持つHMD装置が開発されていますが、それでも体験者の視野をすべてディスプレイで覆うことはできません<sup>3</sup>。視野角の正面、その一部だけにしか画像を提示できないので、その画像が以下にリアルに見えるものであったとしても、そしていわゆる立体視ができていて奥行き方向の位置もきちんと提示される優れた3D画像表示技術が使われていたとしても、見る人にとっては視野の正面に視野の（全体ではなく）一部を使って画像が表示されているわけで、常に画像の範囲、画像の周囲の枠の存在を意識してしまうことになります。

IMAXと呼ばれる巨大なスクリーンで映画などの画像を見たことがある方は多いと思います。IMAXで提示される動画は、それがたとえステレオ画像でなくても、見る人には実在感、その世界に自分が入り込んでいるという感覚を味わうことができます。このような感覚を没入感と呼ばば、IMAXシアターでの体験は、立体視をしているかどうかよりもむしろ視野角の広さの方が没入感の実現には重要であるということを示唆しています。

現在のHMD装置では、IMAXのように広い視野角に同時に画像を表示することができません。HMD装置でステレオ画像を見ると、自分の前方数メートル先にテレビ画面が浮かんでいるかのような感じを受けます。その世界に自分が入り込んでいるという没入感が決定的に欠けてしまっていると少なくとも私には思えます。HMDの持つ視野角の狭さという欠点を補うために、頭全体を覆うフルフェイスヘルメットのようなHMD装置が提案されたこともありました。残念ながら実用的にはなっていません。

広い視野角を確保するための画像表示装置のためにフルフェイスヘルメット型のHMD装置が重すぎるのであれば、いっそのこと巨大なディスプレイに囲まれた部屋のような空間に体ごと入り込んでしまえばいいのではないかと発想が出てきます。実際そのようなVR装置は以前から存在しており、CAVE装置と呼ばれます。CAVE装置はイリノイ大学シカゴ校で1990年代初めに考案されました。オリジナルのCAVE装置は一辺が10フィート（約3メートル）の立方体型で、3つの壁と床面の合計4面がステレオ画像を投影するスクリーンになっていました。壁は半透明のスクリーンになっており、背後に置かれたステレオプロジェクターから立体画像が投影されます。床面には天井に設置されたプロジェクターから立体画像が投影されます。（図2）

各プロジェクターから投影される画像が全然別物であれば、中に入っている人は、4つのスクリーン（3つの壁面と一つの床面）に投影される独立した4つのステレオ画像を見ることになり、特に興味深いものにはならないでしょうが、CAVEでは、それら4つの画像投影するのは、共通した一つコンピュータグラフィクス（CG）の世界で、単に投影される面が異なるだけ、という設定にします。そして、スクリーン同士の境界では、投影される画像同士がきちんとつながって写るように調整します。足下（床面）も含めて広い視野角<sup>4</sup>に映像が投影されるので、IMAX以上の、MHDとは比較にならないほどの高い没入感が作り出されます。

CAVE装置のもう一つの特徴は、ヘッドトラッキングと呼ばれる技術を使って、CAVEの部屋の中に入っている人の視点の位置と視線の角度をリアルタイムで検出していることです。各瞬間、その人の目がCAVEの3次元空間中のどの位置にあり、どの方向を向いているかを測っているわ

<sup>3</sup> しかもそのような特殊なHMDはまだ今のところ大きくて重いという問題もあります。

<sup>4</sup> 正面のスクリーンの方向に頭を向けていれば、水平方向には人間の視野を超える範囲にステレオ画像が投影されず。

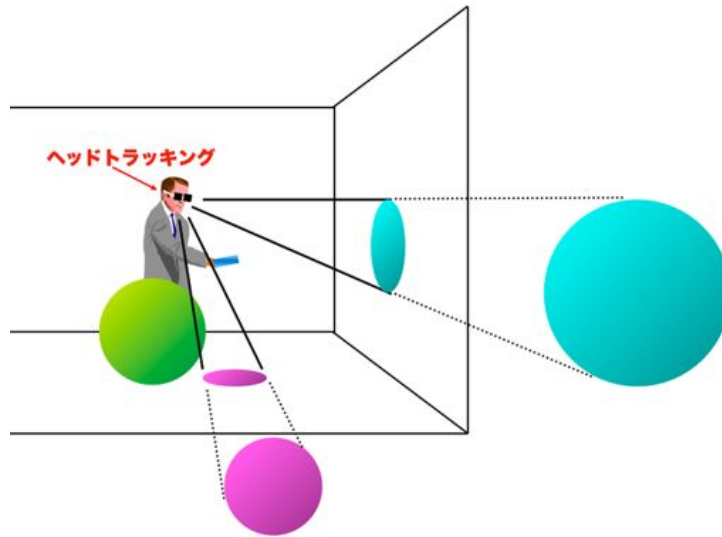


図 2: 没入型 VR 装置 CAVE の概念図。

けです。そして、その人がその位置から CG の人工世界を見たときにどのような画像が見えるべきかをリアルタイムで計算し、それぞれのスクリーンに投影します。ヘッドトラッキングの位置・角度の検出精度は高く、しかも時間分解能も高いので、その人が歩き回っても、しゃがみ込んだりしても、常にその人にとってその位置から見えるべき世界が投影されるので、その人にとっては全てが自然に見えるのです。

CAVE の部屋に入る人は通常ワンド<sup>5</sup>と呼ばれるコントローラを持ちます。このワンドにはボタンやジョイスティックなどがついており、計算機を通じて人工世界と相互作用します。ワンドにもヘッドトラッキングと同じように位置と向きを測る装置が接続しており、たとえば、その人が CAVE の空間の中で、ワンドのボタンを押すことで、その位置に（仮想的な）物体を作り出したり、あるいはその位置に存在する（ように見える）物体をワンドを使って変形したり、動かしたりすることが可能です。

神戸大学では昨年（2011 年）にポートアイランド地区に設置された神戸大学統合研究拠点ビルに新しい CAVE 装置  $\pi$ -CAVE を導入しました（図 3）。 $\pi$ -CAVE 装置は直方体型で、奥行き × 高さ × 幅が 3m × 3m × 7.8m です。我々が知る限りこれは日本で最も大きな CAVE 装置です。

CAVE でなぜ大きさが重要かと言えば、これも視野角に関係します。CAVE でデータを可視化したり、あるいはその可視化が画像を人に見せる場合、CAVE の部屋の中に 2、3 人から、多いときには 5、6 人の人が同時に入ることがよくあります。このようなとき、CAVE の部屋が小さいと、まるでエレベータに多人数が乗ったときのようにお互いの体がスクリーンの画像を覆ってしまいます。CAVE の最大の利点である広い視野角、そしてそれが醸し出す高い没入感がいっぺんに台

<sup>5</sup>magic wand = 魔法の杖



図 3: 神戸大学統合研究拠点に導入された VR 装置  $\pi$ -CAVE。

無しになってしまうのです。しかし  $\pi$ -CAVE ほどの大きさになると 5、6 人が同時に入っても没入感が下がることはありません。

ここで  $\pi$ -CAVE 装置のハードウェアを簡単に紹介しておきます。装置形状は既に述べたとおり  $3\text{m} \times 3\text{m} \times 7.8\text{m}$  の直方体型です。10,500 ルーメンの明るさを持つステレオプロジェクタが全部で 6 台設置されています。解像度は  $1,920 \times 1,200$  ピクセルです。6 台のうち 2 台は正面のスクリーンに画像を投影するために用いており、別の 2 台は床面への投影に用います。この床面用のプロジェクター 2 台は天井に固定されています。残りの 2 台は右の壁面のスクリーンと左の壁面のスクリーンに 1 台ずつ用います。ヘッドトラッキングには赤外線を用いた光学式のモーションキャプチャの技術を使っています。これは映画撮影などでよく用いられるものです。

### 3 VR 技術を用いた可視化例

この章では、 $\pi$ -CAVE 装置を用いて最近我々が行っている可視化の具体例について紹介します。

#### 3.1 地震波伝播シミュレーションの可視化

このシミュレーションは、東大地震研の古村孝志教授によるものです。南海トラフで発生した地震の波が日本列島を伝播していく様子を  $\pi$ -CAVE で 3 次元的に表示します (図 4)。

このシミュレーションでは、直方体領域中の粘弾性体の波動伝播を高精度で解きます。現実の日本列島とその地下構造について観測から得られた物性値の分布を取り込むことで、現実の地震波伝播の複雑な挙動を可能な限り再現するものです。地震発生後、実際の時間で 20 分弱ほどの時間発展のシミュレーションデータを  $\pi$ -CAVE では約 30 秒に圧縮して可視化します。実際のシミュ

レーションは3次元計算ですが、今のところ  $\pi$ -CAVE では地表での揺れという2次元的な分布のみ可視化しています。平野部で長周期の振動が長く残り続ける様子などがリアルに再現されます。

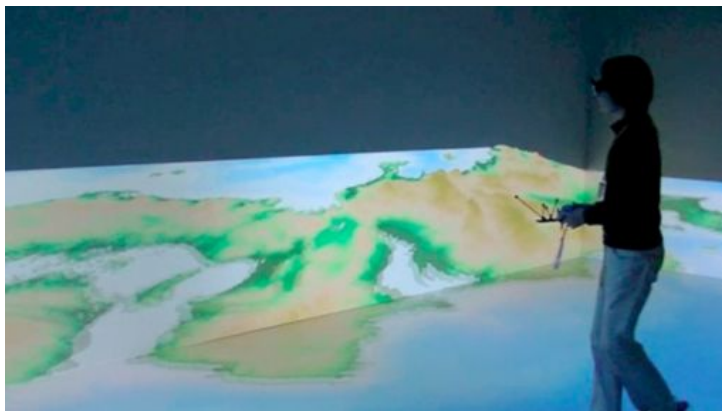


図4: 東大地震研古村孝志教授による地震波伝播シミュレーションの  $\pi$ -CAVE での可視化。

### 3.2 構造物の地震動シミュレーションの可視化

これも地震に関係したシミュレーションですが、上の第3.1章で述べたものとは空間スケールが異なります。このシミュレーションは、東大地震研の堀宗朗教授によるもので、一つの都市のスケールにおいて地震波がどのような影響を与えるかを計算するものです [3, 4]。

地震が遠方で起きた場合、その地震波がある都市にやってきたときに、その都市が受ける被害をある程度予想することは可能です。地震によって受ける都市の被害には様々な種類がありますが、その中でこのシミュレーションが注目するのはビルなどの構造物が受ける影響です。地震波には個性があるので、それぞれのビルがどう揺れるかというのは、到達する地震波によって異なります。建物の揺れ方は、その位置での地盤がどう揺れるか、そしてその建物が力学的にどういう構造をもっているかによって決まります。都市の建築物の力学的な構造はデータベースになっています。地盤がどう揺れるかについては、その都市の地下、地盤の構造が分かっていますので、計算機シミュレーションで予測することができます。そこで、まずある地震波が来たときにその都市の地盤がどう揺れるかを計算するシミュレーションを行い、次にその計算結果を今度は入力データとして、その都市の建物群がどう揺れるかを計算したのがこのシミュレーションです。 $\pi$ -CAVEでの可視化(図5)では、それぞれの建物を構成する頂点の位置をその揺れの大きさに応じて動かすと同時に、揺れの大きさによって色をつけることで分かりやすく可視化しています。

### 3.3 火山噴煙シミュレーションの可視化

このシミュレーションは、東大地震研の鈴木雄治郎博士によって行われたもので、鹿児島県の新燃岳で起きた昨年(2011年)1月の噴火を想定した計算です。このときの噴火では上空の風に



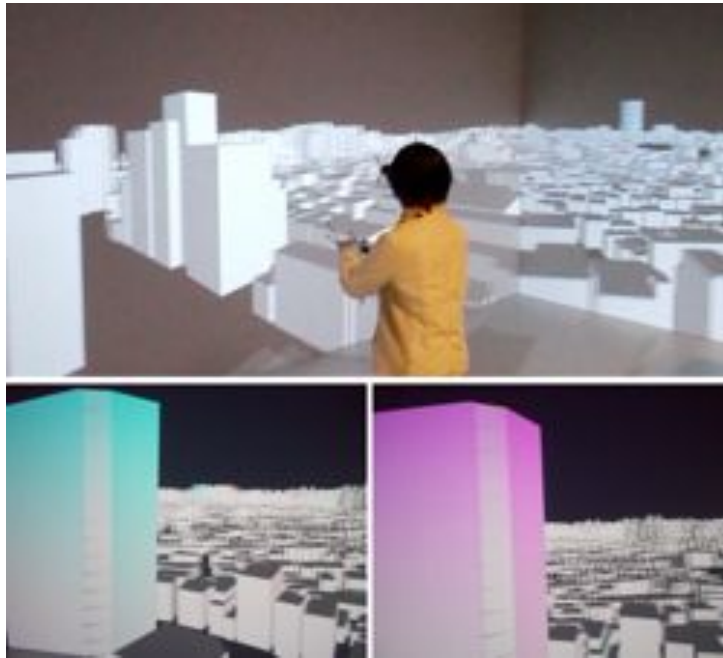


図 5: 東大地震研 堀宗朗教授による都市スケールの地震動シミュレーションの  $\pi$ -CAVE での可視化。

より火山灰が遠方まで運ばれ、堆積した火山灰が各地に被害を与えました。

このシミュレーションは、乱流モデルを取り込んだ一種の流体シミュレーションです。噴火口から噴出した高温の流体が、浮力によって上昇しながら周囲の大気を取り込んでいきます。シミュレーション手法など、詳しくは鈴木博士の論文をご覧ください [5, 6, 7]。このシミュレーションでは、火山灰は流体から力を受ける粒子としてモデル化されています。噴火口の上空の大気の状態、風によって噴煙が流され方が変わります。そして、火山灰粒子の大きさ（そして質量）が異なると違う場所に降着します。 $\pi$ -CAVE での可視化（図 6）では、火山灰データのみを可視化していません。一つの火山灰粒子を一つの球で表示しています。火山灰の大きさは球の色で表現しています。 $\pi$ -CAVE の中に立ち、目の前で噴火する様子を 3 次元的に見るのはかなり迫力のある経験です。特に噴火口の真上から噴火口をのぞき込むとさらに迫力があります。

#### 4 In-Situ 可視化を対話的に見る新しい方法

なぜスーパーコンピュータの演算速度が向上するとポストプロセスとしての可視化が難しくなるのかを以下で説明しましょう。スーパーコンピュータの演算速度は FLOPS という単位で測ります。FLOPS とは一秒間に小数の四則演算が何回できるかという単位です。小数といってもこの場合は、IEEE754 とよばれる規格<sup>6</sup>で決められた倍精度浮動小数点数です。

<sup>6</sup>IEEE754 では、一つの小数を 64 ビットを使って記述します。指数部が 11 ビット、仮数部が 52 ビット、符号が 1 ビットです。10 進数で言えば、これは 15 桁から 16 桁の範囲の小数を記述します。



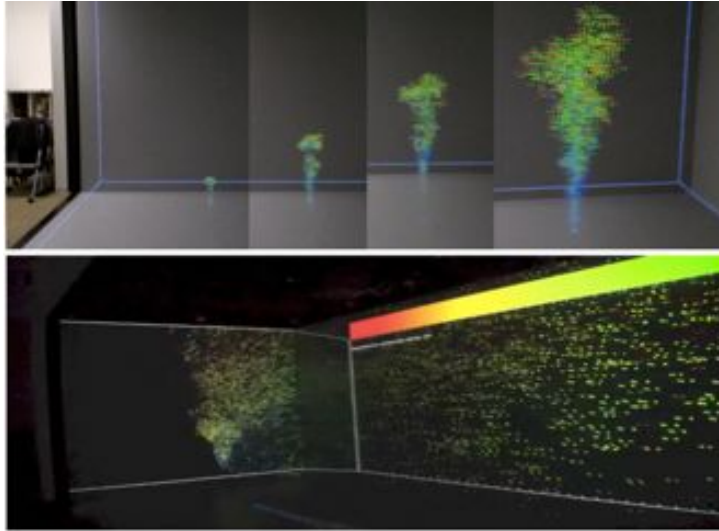


図 6: 東大地震研鈴木雄治郎博士による新燃岳火山噴煙シミュレーションの可視化。火山灰をモデル化した粒子の降着の様子を上空の風がない場合（上図）、風がある場合（下図）の二つの場合について計算した結果を可視化している。

人間が筆算で計算するときの FLOPS 値はいくらでしょうか。円周率を 15 桁で近似して  $\pi = 3.14159265358979$  としたとき、 $\pi^2$  の値を 10 秒で筆算できる人はおそらくいないと思います。100 秒でも難しいでしょう。私の場合はおそらく 1000 秒はかかりそうです。1000 秒で一回の浮動小数点数の演算ができるとすれば、私の FLOPS 性能は 1 mFLOPS（1 ミリフロップス）ということです。最近のパソコンの FLOPS 値はだいたい  $10^{10}$  FLOPS つまり、10 GFLOPS（ギガフロップス）といったところです。最近、世界一位のスーパーコンピュータとなった日本が誇る京コンピュータの演算性能はその名前の由来となった通り 1 京フロップス、即ち  $10^{16}$  FLOPS です。人間の演算性能と比較するのはもう馬鹿らしいくらいなので、パソコンの速度と比較すると京コンピュータは  $10^6$  倍速いわけです。百万倍の速度差のあるものというのは世の中にそうありません。光の速度が  $3 \times 10^8$  m/s で、空気中の音速が約  $3 \times 10^2$  m/s ですから、その速度比が  $10^6$  です。

京コンピュータをはじめとし、世界最速のスーパーコンピュータの演算性能は、時間の関数として指数関数的に伸びています。これは集積回路上のトランジスタの数の向上速度について述べられたムーアの法則<sup>7</sup>の一つの帰結です。このような指数関数的な向上が、最初のスーパーコンピュータ（Cray-1）の登場以来、もう 36 年間も続いていて、これがまだしばらくは続きそうです。科学技術の世界では、あるいは世の中のどんなことにせよ、何かの性能が時間の関数として指数関数的に向上し続けるような時期がこれほど長続きすることは珍しいことだと思います。それでもこのような時期もいつかは終わりを告げるでしょうから、この意味で今、我々は実に特別な時代に生きているのだと思います。

浮動小数点数の四則演算を処理するプロセッサの演算性能が指数関数的に進歩している今の時

<sup>7</sup>18ヶ月で2倍

代、計算に関連する他の技術もまた指数関数的な進歩をしていることが望ましいのは言うまでもありません。そうでなければ、プロセッサ技術とその技術との格差が指数関数的に開いていきます。そうすると、その両者のアンバランスが極めて深刻な問題になっていきます。

実際、演算プロセッサの進歩に追いついていない重要技術は存在します。一つは、ネットワークの転送性能です。

ネットワークの帯域も時代と共に順調に進歩を続けていますが、演算性能の爆発的とも言っているほどの指数関数的な進歩には追いついていません。もう一つ、演算性能の進歩に追いついていない技術はハードディスクへの転送性能です。現在のスーパーコンピュータでは、計算結果のデータは最終的にはそのスーパーコンピュータシステムに接続されたハードディスク装置に保存されます。この点はPCと同じです。ところが、ハードディスク装置にデータを書き込むための性能がやはり、演算性能の進歩に追いついていないのです。

ネットワーク帯域とディスク入出力の帯域という二つの性能が計算機シミュレーション研究におけるボトルネックになっており、その状況は今後ますます深刻になっていくことが予想されます。ポストプロセスで可視化をするためには、計算結果としての数値データを保存しなければなりませんから、このボトルネックは深刻です。

そこで、近年スーパーコンピュータで計算を行いながら可視化を行う In-Situ 可視化（その場可視化）と呼ばれる手法が注目を集めています。In-Situ 可視化では計算機シミュレーションの進行と同時に可視化も行います。ポストプロセスのための数値データを出力する代わりに、可視化結果としての画像データを出力します。画像は2次元なので、3次元の数値データよりもデータサイズは相対的に小さくて済みます。

ただし、単純な In-Situ 可視化では、可視化を行うときの視点の設定や、採用する可視化手法やその内部パラメータなどの設定を誤るともう一度計算を行わなければいけないという問題があります。興味深い現象、そのシミュレーションで理解したい現象がどこでどのように起きるかをあらかじめ分かっているのであれば問題ありません。可視化を行うにあたっては、一般的なコンピュータグラフィックス処理と同様に、カメラの位置を決め視野を設定する必要がありますが、興味深い現象が起きる場所がその視野の中に入っているように設定し、その現象の本質がとらえられる可視化手法で現象を（シミュレーションの進行に合わせて）可視化処理して画像を生成していけば良いからです。問題は、あらかじめどこで興味深い現象が起きるのか予測がつかない場合です。そのような場合、In-Situ 可視化を行いながら計算機シミュレーションをしても、面白い現象が可視化のカメラに捕らえられていなければ、もう一度計算を行わなければなりません。

このような問題を解決する上で Steering シミュレーションという方法が提唱されています。この Steering シミュレーションでは、計算機シミュレーションを行いながら、その時に計算機の中で再現されている現象を In-Situ 可視化で画像化します。ここまでは通常の In-Situ 可視化と同様です。そしてその可視化画像をシミュレーションを実行する研究者が手元の PC 等で観察しながら、必要があれば、たとえば興味深い現象がカメラの視野に捉えられていないときなど、カメラの位置や視野の設定、あるいは可視化手法などをリアルタイムで変更する方法です。Steering シミュレーションでは、可視化に関わる量だけでなく、シミュレーション対象の現象に関わるパラメータもリアルタイムで変更します。計算している現象を見ながら、その現象そのものに手を加えるわけです。

舵取りをしながらのシミュレーションという意味で **Steering** シミュレーションと呼ばれます。

シミュレーション対象を見ながら自由自在にそれをコントロールできるこの **Steering** シミュレーションはある意味で究極の計算機シミュレーションスタイルと言えるかもしれません。しかしながら現在のスーパーコンピュータでは、バッチジョブ方式<sup>8</sup>で計算を行うのが普通なので、完全な **Steering** シミュレーションは現実的ではありません。ジョブを待ち行列に放り込んでから、いつその計算が実行されるか分からない上、一度動き始めた計算ジョブを外部からダイナミックにコントロールする手段が提供されていないからです。

そこで我々は最近、バッチジョブ方式でスーパーコンピュータに投入する計算機シミュレーションに対しても **In-Situ** 可視化が有効な方法を提案しました [8]。それは簡単に言えば、ユーザが必要とするかもしれないあらゆる可視化画像を全て **In-Situ** で可視化し、画像として出力するという方法です。このシミュレーションスタイルにおける「計算結果」とは数値データではなく、大量の動画ファイルです。カメラの位置を 1000 点とし、それぞれのカメラから 1000 通りの可視化を行えば、一度の計算機シミュレーションで  $1000 \times 1000$ 、つまり 100 万個の可視化動画ファイルが生成されます。一つの動画ファイルがデータ圧縮の結果、10 MB のサイズになっているとすれば、100 万個の動画ファイルといえども「わずか」10 TB にしかありません。10 TB というと大きいように感じるかもしれませんが、現在のスーパーコンピュータでも少し大きめのシミュレーションをすれば、ポストプロセスとしての可視化を行うために 3 次元数値データを十分細かい時間間隔で出力すると、この程度のデータサイズは優に超えます<sup>9</sup>。演算性能の指数関数的な進歩を考慮すれば、ポストプロセスのための出力データの大きさはますます増大するのは目に見えていますから、10 TB の動画ファイルはそれに比べれば圧倒的に小さいということになっていくでしょう。

我々が提案している新しい可視化スタイル、**IVI-SV** (**I**nteractive **V**iewing of **I**n-**S**itu **V**isualization) では、大量の動画ファイルが出力されます。計算機シミュレーションの中で何か興味深い現象が再現されていたとしたら、この膨大な動画ファイル群の中にその映像が映っているはずですが、とはいえ、動画ファイル群がそのままではあるだけでは、それを引き出すことができません。我々は動画ファイル群から適切な静止画像列を対話的に抜き出すプレイヤーを開発しました。このプレイヤーの入力データは  $N$  個の動画ファイルです。それを **Movie(1)**, **Movie(2)**, **Movie(3)**, ..., **Movie( $N$ )** とします。それぞれの動画は、 $M$  個の連番静止画像から構成されているとします。たとえば  $i$  番の動画ファイル **Movie( $i$ )** は **Movie( $i$ ):Image(1)**, **Movie( $i$ ):Image(2)**, ..., **Movie( $i$ ):Image( $M$ )** で構成されています。このプレイヤーの出力は PC のウィンドウ上に表示される動画です。アニメーションとして再生されるので、時間方向の整数パラメータ  $j$  を 1 から  $M$  まで上げながら対応する静止画を画面に表示するわけですが、通常の動画プレイヤーソフトとは異なり、我々のプレイヤーは、どれか特定の動画ファイル (パラメータ  $i$ ) を固定したものではなく、ユーザが対話的にこの  $i$  を対話的に変更することが可能で、このとき同時に  $j$  も上がっていきます (つまり動画として再生さ

<sup>8</sup>一つの計算に関わる作業をあらかじめファイルに記述し、それを計算機の待ち行列に登録する方式。実際にいつ計算が実行されるかは待ち行列の混み具合などに依存する。

<sup>9</sup>たとえば流体計算で一辺が 3000 格子点の 3 次元シミュレーション (全格子点数  $3000^3$ ) を考えます。ポストプロセスで 100 秒程度の動画をつくるためには、30 フレーム毎秒として 3000 個の時間スナップショットデータが必要です。一格子点あたり、速度 3 成分と圧力の 4 つの物理量は最低でも出力する必要があります。それぞれの物理量を単精度浮動小数点数 (4 B) 数値データとして保存すると、出力データは全部で、 $3000^3$  格子点  $\times$  3000 時間ステップ  $\times$  4 物理量  $\times$  4 B、つまり約  $10^{15}$  B (1 PB) の数値データを出力しなければなりません。

れています)。

この IVI-SV 法の具体的な適用方法は例えば次のようなものです。

計算機シミュレーションのシミュレーション領域全体を覆うように、シミュレーション領域の内外に  $N$  台のカメラを設置します。そのカメラからそれぞれ  $L$  通りの可視化手法で対象を可視化します。そして、全部で  $N \times L$  種類の In-Situ 可視化をしながらシミュレーション計算を実行します。これはバッチジョブで構いません。計算結果は  $N \times L$  個の動画ファイルです。その動画ファイルをパソコンに転送し、我々が開発したプレイヤーに読み込ませて再生を始めます。最初はデフォルトの動画ファイル  $i_0$  が再生されていますが、キーボードやマウスを通じて異なる  $i$  にめらかに切り替えます。同じ現象を別の角度から撮ったカメラが複数台あり、その角度の差が十分に小さければ、現象を（通常のポストプロセス可視化の可視化ソフトで通常行うように）ぐるっと回りながら観察しているかのような動画の再生方法が可能です。カメラ位置に限らず、他の可視化パラメータについてもダイナミックな変更が可能です。たとえば等値面表示による可視化を考えます。ある等値面のレベルから、別の等値面のレベルに切り替えるとき、ポストプロセスでの可視化では、ユーザが等値面のレベルを変更した後に、等値面を構成する可視化アルゴリズムを実行するわけですが、IVI-SV 可視化スタイルでは、変更後の等値面レベルで In-Situ 可視化した動画ファイルから静止面を取り出すように切り替えるだけです。つまり、この方法により（十分な数の動画ファイルさえあれば）ポストプロセス可視化と本質的に同じ可視化解析が可能となるわけです。

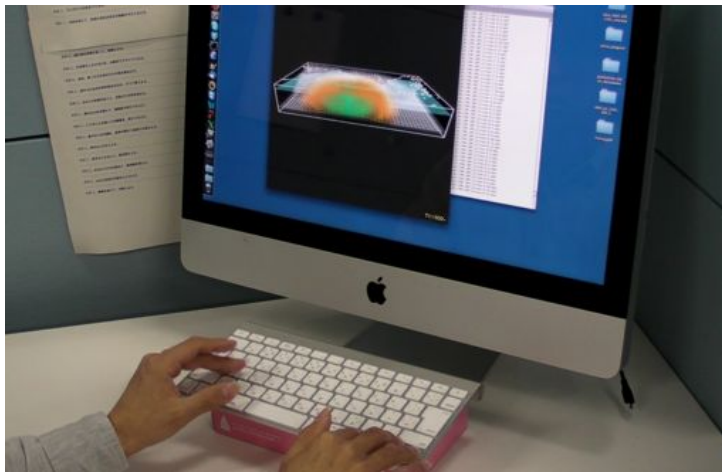


図 7: IVI-SV 法で作成した 390 個の動画ファイルを対話的に解析している様子。

我々は IVI-SV 法の有効性を実証するための検証実験を行いました (図 7)。対象とするシミュレーションは 3.2 章で述べた同様に、古村孝志教授の地震波伝播シミュレーションです。この実証実験で用いたシミュレーションコードと可視化プログラムも古村教授から提供されたものです [9, 10]。この実験ではカメラの数は  $N = 130$ 、可視化種類の数は  $L = 3$ 、時刻ステップの数は  $M = 60$  としました。計算は神戸大学に設置された PC クラスタで行い、40 個の MPI プロセスで並列計算を行いつつ、In-Situ 可視化を行いました。出力データは、390 個 ( $=N \times L$ ) の動画ファイルで、合計の

サイズは 57 GB でした。なお、カメラの配置は 2 次元的とし、計算領域の中心からの半径一定の球面上に置きました。球面上の  $N$  個のカメラをできるだけ一様に分布させることはそれほど単純な問題ではありません。今回はインヤン格子の格子点上に置きました。インの格子点上に 65 点、ヤンの格子点上に 65 点です。

今回開発したプレイヤーはこの 390 個の動画ファイルを読み込み、問題なく再生させることができました。キーボードで矢印キーを押すと、その方向に置かれた隣のカメラから撮影した可視化画像が再生されます。動画は瞬時に反応しますので、何も知らされないユーザはそれが通常の (ポストプロセスの) 可視化であると思ひ込むでしょう。この検証実験の結果、この IVI-SV 法は十分実用的であるだけでなく、極めて効果的な可視化手法であることを確認することができました。

## 5 まとめ

本論文では、シミュレーションデータの可視化に関して最近の我々が行っている二種類の取り組みについて紹介しました。一つは没入型と呼ばれる方式のバーチャルリアリティ (VR) 技術を用いてデータを 3 次元的に可視化する方法、もう一つは IVI-SV 法という我々が新しく提案している方法です。

VR 技術を可視化に応用する取り組みは長い歴史があります。そのような中、第 3 章で紹介した我々の VR 可視化の取り組みは、可視化手法そのものに我々の独自性があるわけではありません。しかし、最近のスーパーコンピュータの進歩に伴って、計算される現象が複雑になっているため、可視化すべきデータが大規模、複雑になっています。VR 環境を使った可視化で重要なのは、画像を生成する速度です。CAVE 型の VR 装置では、CAVE の部屋の中を観察者が自由に歩き回ります。その観察者にとって常に自然な仮想現実世界を提示するためには、1 秒間に 30 回程度、コンピュータグラフィックスの画像を生成する必要があります。ヘッドトラッキング装置によって測定した観察者の視点の位置から、CAVE のスクリーンに投影すべき画像を毎秒 30 フレーム程度の高速でレンダリングするためには様々な技術的工夫が必要となります。

この論文では紹介しませんでした。VR 技術を用いた可視化に関して、我々が独自の手法を考案した研究もあります。例えば、流れに凍り付いた磁力線の可視化手法 [11] が最近の例です。

我々は VFIVE という名前の汎用可視化ソフトウェアも開発しています [12]。VFIVE は CAVE 型 VR 装置でスカラー場やベクトル場を可視化するための汎用の可視化ソフトです。VFIVE は現在、核融合科学研究所、中央大学、海洋研究開発機構、神戸大などで研究や開発に活用されています [13]。

CAVE 型 VR 装置で複数のアプリケーションを使う場合、いちいち CAVE の部屋の外に出て、コマンドを打ち込み、CAVE の部屋に入ってアプリケーションを使うということを繰り返す必要があります。これは OS のないパソコンのようなもので、実用上大変不便です。そこで我々は CAVE 用のアプリケーションランチャを開発しました [14]。これはパソコンのデスクトップ環境に相当するものです。CAVE の空間に浮かぶ 3 次元タッチパネルを手を持ったコントローラ (ワンド) で触って選択することで、対応するアプリケーションを立ち上げることができます。

本論文では VR 技術を使った可視化に加えて、IVI-SV 法という我々が提唱している新しい可視

化手法についても紹介しました。スーパーコンピュータの演算性能の指数関数的な進歩は今後も続くことが予想されます。データ可視化に関しても従来のようなポストプロセスを前提とした可視化スタイルは維持できなくなるでしょう。ネットワーク帯域とディスク入出力帯域がボトルネックになるからです。また、シミュレーションが対象とする現象が複雑になるにつれ、それを可視化するための処理も複雑になってきており、既に並列プロセッサをつかって可視化処理することは当たり前になっています。将来は、大規模並列計算機上でシミュレーション計算すると同時に、可視化処理も進めていくという In-Situ 可視化が不可避となるでしょう。我々が提案する IVI-SV 法はバッチジョブで投入した計算機シミュレーション結果を In-Situ 可視化し、それをポストプロセスで対話的に解析するという方法です。実証実験の結果から、我々はこの方法が大変有望な方法であると考えています。

## 謝辞

第 3.1 章で紹介した地震波伝播シミュレーションデータは、東大地震研の古村孝志教授から提供されたものです。第 3.2 章で紹介した都市の地震動シミュレーションデータは、東大地震研の堀 宗朗教授から提供されました。このデータの  $\pi$ -CAVE での可視化プログラムは神戸大学大学院生の守本敏明君が開発しました。第 3.3 章で紹介した火山噴煙シミュレーションのデータは、東大地震研の鈴木雄治郎博士から提供されたものです。第 4 章で紹介した IVI-SV 法の実証実験シミュレーションで用いた地震波伝播シミュレーションのシミュレーションコードと可視化プログラムは東大地震研の古村孝志教授から提供されたものです。図 7 に示した IVI-SV 法の動画プレイヤーは神戸大学大学院生の山田知輝君が開発しました。 $\pi$ -CAVE での可視化プログラムの開発にあたっては富山麻子博士の協力をいただきました。本研究は JSPS 科研費 (23340128)、および高橋産業経済研究財団から助成を受けました。

## 参考文献

- [1] 陰山聡, 宮腰剛広. 低エクマン数領域における地球ダイナモシミュレーション. 九州大学応用力学研究所研究集会報告, 20ME-S6, pp.69–78, 2009.
- [2] Ivan E. Sutherland. A head-mounted three dimensional display. *Proceedings of Fall Joint Computer Conference, part I*, pp.757–764, 1968.
- [3] T. Ichimura and M. Hori. Structural seismic response analysis based on multiscale approach of computing fault-structure system. *Earthquake Engineering & Structural Dynamics*, vol.38, pp.439–455, 2009.
- [4] Muneo Hori, Kenji Oguni, and Tsuyoshi Ichimura. Integrated Simulation For Earthquake Hazard and Disaster Prediction. *Journal of Earthquake and Tsunami*, vol.03, pp.121–141, 2009.
- [5] Takehiro Koyaguchi, Yujiro J. Suzuki, and Tomofumi Kozono. Effects of the crater on eruption column dynamics. *Journal of Geophysical Research*, vol.115(B7):B07205, 2010.



- [6] Y. J. Suzuki and T. Koyaguchi. Numerical determination of the efficiency of entrainment in volcanic eruption columns. *Geophysical Research Letters*, vol.37(5):L05302, 2010.
- [7] Yujiro J. Suzuki and Takehiro Koyaguchi. 3-D numerical simulations of eruption column collapse: Effects of vent size on pressure-balanced jet/plumes. *Journal of Volcanology and Geothermal Research*, vol.37, vol.221-222, pp.1–13, 2012.
- [8] Akira Kageyama and Tomoki Yamada. An Approach to Exascale Visualization: Interactive Viewing of In-Situ Visualization. *submitted*, 2013.
- [9] Takeshi Furumura and L. Chen. Large Scale Parallel Simulation and Visualization of 3D Seismic Wavefield Using the Earth Simulator. *Computer Modeling of Engeneering and Sciences*, vol.6, pp.153–168, 2004.
- [10] S. Takemura, T. Furumura, and T. Saito. Distortion of the apparent S -wave radiation pattern in the high-frequency wavefield: Tottori-Ken Seibu, Japan, earthquake of 2000. *Geophysical Journal International*, vol.178, pp.950–961, 2009.
- [11] Kaori Murata and Akira Kageyama. Virtual Reality Visualization of Frozen-in Vector Fields. *Plasma and Fusion Research*, vol.6:2406023–1–5, 2011.
- [12] Akira Kageyama and Nobuaki Ohno. Immersive VR Visualizations by VFIVE. Part 1: Development. *International Journal of Modeling, Simulation, and Scientific Computing*, vol.4, 2013, in press.
- [13] Akira Kageyama, Nobuaki Ohno, Shintaro Kawahara, Kazuo Kashiyama, and Hiroaki Ohtani. Immersive VR Visualizations by VFIVE. Part 2: Applications. *International Journal of Modeling, Simulation, and Scientific Computing*, vol.4, 2013, in press.
- [14] Akira Kageyama and Youhei Masada. 3-D Desktop Environment and Applications for CAVE-type Virtual Reality Systems. *Journal of Physics: Conference Series*, 2013, in press.