

# 国立天文台 4D2U プロジェクトにおけるシミュレーションデータの 可視化ツールの開発について

武田 隆顕\*

## “Zindaiji” and “Oosawa”, applications for visualizing particle data and volume data

Takaaki TAKEDA\*

### Abstract

Visualizing 3-dimensional or 4-dimensional data is important in both aspects of tools for research and tools for public outreach. We have developed applications for visualization of particle simulation data and volume data. Here, we report the achievement of these applications, and describe the merit and the demerit of developing such home-brewed visualization tools.

**Keyword:** Visualization

### 概要

3次元、または4次元（3次元+時系列）のデータを可視化することは、研究を推進する手段としても重要であり、パブリックアウトリーチの手段として重要である。国立天文台 4D2U プロジェクトでは、粒子データを可視化するツールおよびボリュームデータを可視化するツールを開発し、それによる可視化映像の作成と公開を行なっている。その成果と、このような可視化のための自家製ツール開発の利点およびにその限界について記述する。

### 1. はじめに

国立天文台 4D2U プロジェクトは、最新の天文学の成果を可視化するためのプロジェクトである。正式名称は4次元デジタル宇宙プロジェクトである<sup>1)</sup>。英語では 4-Dimensional Digital Universe であり、アクロニム（頭文字）を使って 4D2U と読んでいる。4D2U にはまた、“4-D to you”（4次元をあなたに）という意味も込められている。ここでの「4次元」は宇宙の次元（空間3次元と時間1次元）を意味している。プロジェクトの活動の一環として、PC を使って宇宙のスケールを理解することのできるビューワーアプリケーションや、天文シミュレーションを元にした可視化映像の製作を行い、ウェブ公開やシアターにおける公開を行っている。

数値シミュレーションの可視化には、一般的に大きく分けて2つの目的が存在する。一つは研究者自身が直感的に把握しやすい形で計算結果を見るために行われる、研究の補助ツールとしての可視化である。こうした可視化には、見た目の綺麗な画像や映像を作成することよりも、物理量が把握しやすく、簡単に視点の移動などの操作ができるといった素早い応答性が最も重要な要素になる。もう一つは、研究の結果を単なる点や線による図やグラフではなく、専門外の人間に対しても理解しやすく印象的な画像、映像を作成するための可視化、つまり「見せる」ための可視化である。後者の可視化は、研究成果を社会に還元するパブリックアウトリーチという意味でも重要である。本稿では主に後者に重きを置いた、4D2U プロジェクトでの大規模天文シミュレーションの可視化事例を紹介する

---

\* 国立天文台天文シミュレーションプロジェクト (Center for Computational Astrophysics, National Astronomical Observatory of Japan)

セクション2では我々のプロジェクトでの粒子シミュレーション可視化の詳細について述べる。また、セクション3では流体シミュレーションの可視化について述べ、セクション4ではまとめと今後の進展について述べる。

## 2. 粒子シミュレーション可視化ツール開発

粒子シミュレーションは、天文学の分野では非常に多くみられるシミュレーションである。なぜなら、星団内の星の運動や、微惑星からの惑星の形成など、天文学のトピックの多くで質点とみなせるような粒子同士の重力相互作用が重要な役割を果たすためである。4D2Uプロジェクトではそのような粒子シミュレーションをもとに、多くの公開用の映像を作成<sup>2)</sup>している(図1)。

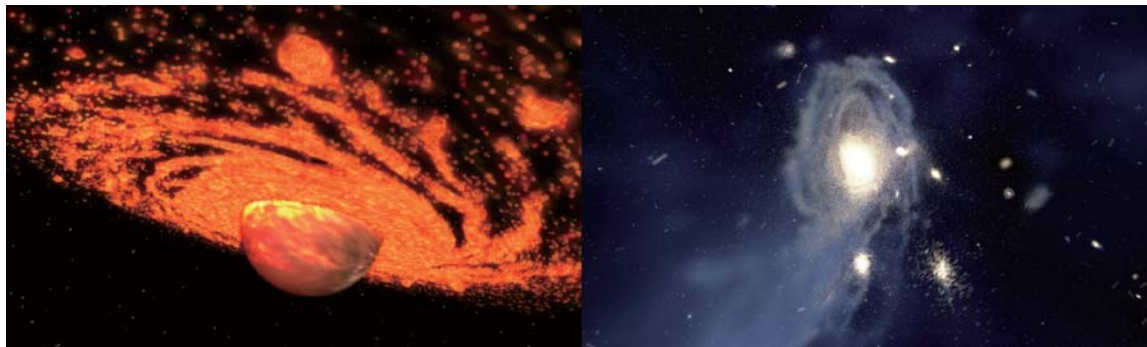


図1. 左)「月の形成」のシーン。巨大衝突によって原始地球周りに撒き散らされた月の材料が集積する過程を示している<sup>3)</sup>。右)「渦巻銀河の形成」のシーン。星をあらわすスーパーパーティクルとガスをあらわすSPH粒子を表示している。星とガスに100万粒子を用いたシミュレーションの可視化映像である<sup>4)</sup>。

我々は、粒子系シミュレーションの可視化映像を製作するにあたり独自の可視化ツールの開発を行い、主にこのツールを使って映像製作を行った。これは、独自開発ツールであればシアター公開のためのカスタマイズや、付加的な映像機能の追加を行いやすいという判断で開発を開始したものである。このツール(Zindaiji)はC++で開発され、32bit Windows環境で動作するものとなっている(図2)。Zindaijiは、天文カタログを基にした宇宙のビュープログラムであるMitaka<sup>5)</sup>など4D2Uプロジェクトで開発された他のアプリケーションと同様にソースコードとともにウェブサイトで公開している<sup>6)</sup>。

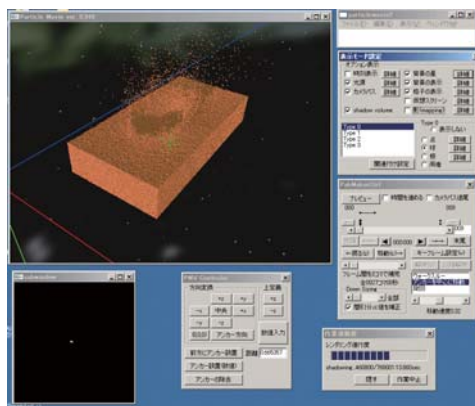


図2. Zindaiji 実行画面。

天文分野では、多数の粒子からなる重力多体計算などが数多く行なわれている。一般的な用途の映像製作ツールでは、そのようなデータの取り扱いが難しい。一般的なCG製作ツールは、多くの機能を有する反面、消費するメモリの量や描画速度、読み取り速度の点で大規模シミュレーションを直接可視化するのが難しいことが多いためである。

粒子系シミュレーションの可視化で最も基本的な機能は、各スナップショットデータにおける粒子位置を表示することである。しかし、通常はシミュレーション結果の出力をそのまま動画にすると映像としては時間間隔が荒すぎるため、スナップショット間の補間が必要になる。我々は、足りないフレームの情報を補うために、2つの時間ステップの位置と速度情報

から、三次関数でスナップショット間を補間するエルミート補間を実装した。

大規模データの可視化にあたっては、描画速度とともにメモリ消費量が大きくなる問題になる。通常のCG製作のようにオブジェクト毎にさまざまな描画情報を持つことはせずに、描画に必要な不可欠なデータのみを保持し、その他の情報はグループごとにまとめて持つことでメモリの節約を行っている。粒子の位置情報は時間についての三次関数で情報を保持しているため、1ステップ1粒子当たり、位置、速度及びに加速度、加加速度の3次元情報が必要となる。float形式でこの情報を持つ場合、粒子当たり48 byteが必要である。通常は、シミュレーションの種類に応じて粒子半径や粒子の種類、温度や粒子番号などの情報が必要であり60byte程度のメモリを消費する。そのため、百万粒子を用いたデータを保持するために1ステップで60MBほどのメモリを消費することになる。

こうした時系列データを扱うためには、必要に応じたメモリ管理を行なうことが重要になる。つまり、HDDから読み込んだデータはメモリに保持しておき、時間軸の移動に対するレスポンスを良くすると同時に、システムのメモリ領域を使い切ってしまうために、必要性が薄くなったデータを順次メモリから開放する必要がある。

こうしたメモリ管理機能が必要なアプリケーションの良い例として、映像編集ソフトウェアがあげられる。Adobe After Effectsなどの映像編集ソフトでは、何度も同じシーンを再生しながらタイミングや映像効果の調整を行なう。その際に、一度表示された画像情報はメモリに確保し、編集時に利用される際のレスポンスの向上を行なっている。多くの場合、映像のシーン全体はメモリ上に乗らないために、以前の表示から時間の経ったフレームの情報を順次メモリから開放することで、使用メモリが一定値内に収まるように実装されている。Zindaijiにおいては、メモリ上にデータを保持しつつ、メモリの使用量が上限に近づいた場合には、あるフレームが現在表示しているフレームからどれだけ離れているのかと、以前に表示されてからの時間を重みをつけて足し合わせ、もっとも大きい値のフレームの情報から順次メモリの開放をするという実装を行なった。

また、映像編集ソフトウェアでは、レイヤーの位置や透明度といった情報を、タイムライン上にキーフレームを配置することで時間変化を自由に編集することができる。Zindaijiにおいても、カメラワークを設定するためのカメラ位置などを、キーフレーム編集を行なう機能を実装した(図3)。Zindaijiにおいては、カメラ位置など、最小限のパラメータのみキーフレーム編集を行なえるといった単純な実装であったが、このようなタイムライン編集は、映像製作に当たっては非常に重要である。Zindaijiの経験を元に、次に開発を行なった流体データ用の可視化ツールでは、大幅にタイムライン機能の充実を図っている。



図3. Zindaijiで実装したタイムライン。メモリ上にデータのある領域は緑の線で表示している。

これらのメモリ管理機能とカメラワークの編集機能を土台として、実際にデータを画面上に表示し、映像を出力するには、グラフィックボードによるポリゴンの表示機能を用いている。土星のリング粒子のシミュレーションなどのように、物理的な半径を持った粒子の場合には多角形で近似した球(図4)を表示しており、SPHによる粒子法による流体シミュレーションにおけるガスの表示や、星団における星などの実質的に大きさを持たない光点の表示には、ガスや光る点のテクスチャを適用したポリゴンを、カメラの方向を向いた状態で奥から順番に表示することで表示を行なっている(図5)。これらは表示方法としては標準的な技法であるが、可視化映像は大規模なシミュレーションを元にリアルな物理運動を表示することから、非常に印象的な映像を作成することも可能である。

こうしたポリゴン表示の品質をあげるために、Zindaijiには影の表示の実装を追加している。実装した手法の一つはシャドウボリューム法<sup>7)</sup>である。図6の模式図に示したように光源方向から伸びる透明な影ポリゴンによって影の領域を決定する。最終的な表示を作成するには、物体の手前に影ポリゴンの表面が何枚あるかを判定することで、影をあらゆる領域の内部にあるのか外部にあるのかを判定することができる(ステンシルシャドウボリューム法)。この方法は、図4のように大きさのある粒子同士で落とす影の作成に適用することができ、質感を向上させることができる。

一方、ガス粒子同士で落ちる影は、シャドウマッピング法<sup>8)</sup>に似た方法で実装している。描画を2回実行し、最初に光源から見た表示を行なうことで、個々のガス粒子がどれだけ光源から隠蔽されているかの度合いを得て、それに基づいて最終的な表示を行なう方法である。最初の段階の表示では、光源に近い粒子から順に表示を行い、粒子の描画をする前に画面上でのピクセルの色を取得する。これにより各粒子が手前の粒子によって光源からどれだけ隠蔽されているかの度合いを得

て、それをもとに実際の表示の際に影をつける実装となっている。この方法は、光源からの隠蔽の度合いの情報を粒子ごとにしか得ることができないために、多くのガス粒子から構成される物体でなければ十分な解像度を得ることができないという欠点があるが、多数の SPH 粒子を使った天体の衝突シミュレーションなどには十分に適用することができる (図 6)。

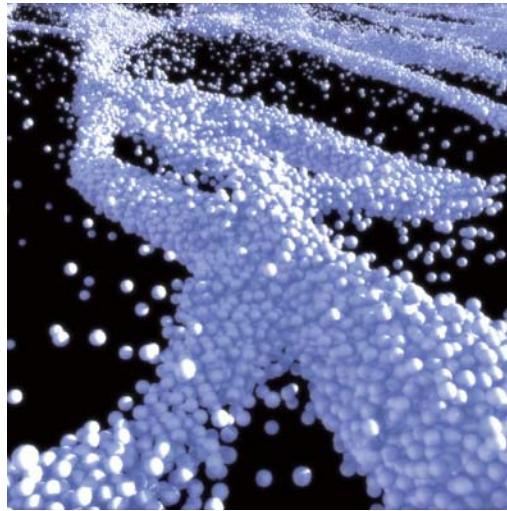


図 4. 「土星リングのダイナミクス」より球の集合体としての粒子シミュレーション表示。

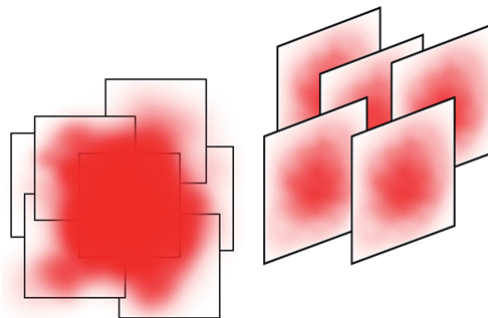


図 5. SPH 粒子などのガス粒子の表示方法の模式図。半透明な板を透明度に応じて奥から表示することで、ガスの分布を表現する。

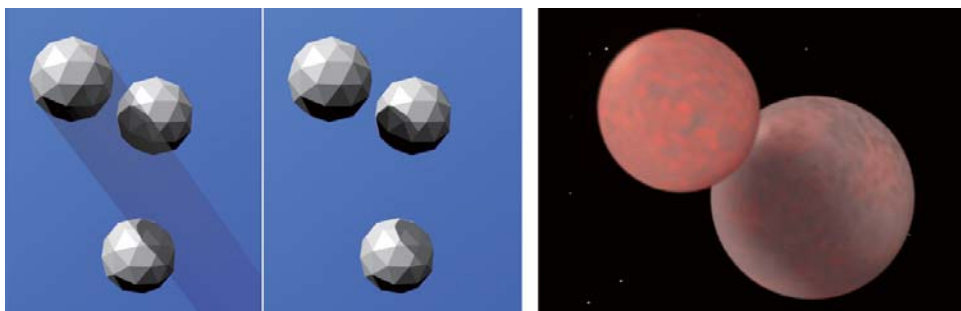


図 6. 左) シャドウボリューム法の模式図。右) SPH 粒子による天体同士の影の表現 (「月の形成」より)。

Zindaiji のように、あるタイプのデータに特化した自家製のツールを開発することで、通常的手段で映像化の難しい大規模なデータを、ある程度高い品質での映像化することが可能となった。特に「渦巻銀河の形成」などは、SIGGRAPH2007 において Electronic Theater で上映されるなど国際的にも高い評価を得ることができた。しかし、Zindaiji を開発およびそれを利用しての反省点として、以下のようなものがあげられる。

自家製ツールの限界として、レンダリング機能に制約があり、ある程度以上の綺麗さを持った映像をつくりにくく、GUI の操作性の完成度にも問題がある。また、本当に大規模なデータを可視化するためには、階層化されたデータへの対応といった機能が必要になるが、そうした点に対応していない。

これらの問題に対応した粒子データの可視化用ツール開発は今後の大きな課題である。現在、以上の経験をふまえた上で、次のセクションで述べる流体データの可視化のためのコードの開発を行なっている。

### 3. 流体シミュレーション可視化ツール開発

4D2U プロジェクトにおける可視化映像は、粒子シミュレーションの可視化映像が多く、その他に多くのシミュレーションが行なわれている流体シミュレーションの可視化が少ないという問題があった。現在流体などのボリュームデータから、一般公開などで用いるための綺麗な映像を作成するために、フリーウェアでレイトレーサー機能を持つ PovRay<sup>9)</sup> を用いてレンダリングを行なう GUI ツールの開発を行なっている (図 7)。これは、グラフィックボードを使った高速なプレビュー表示画面を見ながら GUI でカメラパスの設定といった操作を行い、最終的な出力をレイトレースを用いて行なうツールになっている。最終出力を PovRay によって行なうことは、Zindaiji で見られたような、自家製ツールではある程度以上の綺麗さを持った映像を作ることが難しいという欠点を補うことが狙いである。このツールも国立天文台のある地名から Oosawa と名づけた。C++ で開発され、Windows および Linux 環境でコンパイル可能なマルチプラットフォームのアプリケーションとして公開を行なっている<sup>10)</sup>。

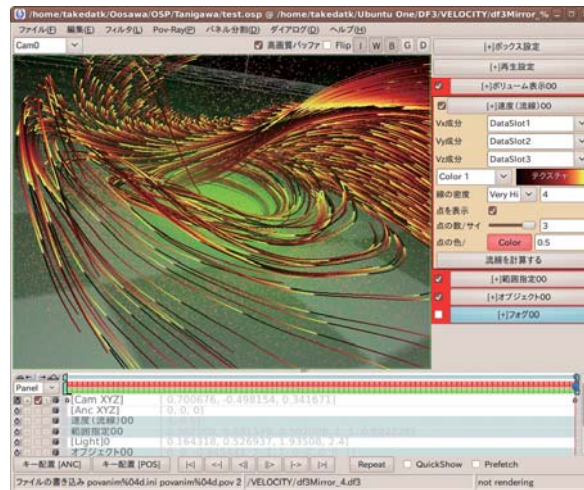


図 7. Oosawa 実行画面。ガスと流線を表示している。

時系列に沿って進化するデータは、そのデータ量はスナップショットの数だけ増えてゆくために膨大になりがちである。Oosawa においてもメモリの管理が重要であるため、映像編集ソフトのタイムライン編集のユーザーインターフェイスを大きく参考としている。Oosawa では時系列データの読み込み時のメモリの確保と開放を基本として、カメラの位置のほか、光源位置、等値面の閾値や断面図のスライス面といった主だったパラメータに関して、タイムラインにキーフレームを配置して編集を行なえるように実装を行なっている (図 8)。このような形式を取ることで、長い時系列データやその時間進化に応じたカメラワーク設定を高い自由度で行なうことができる。

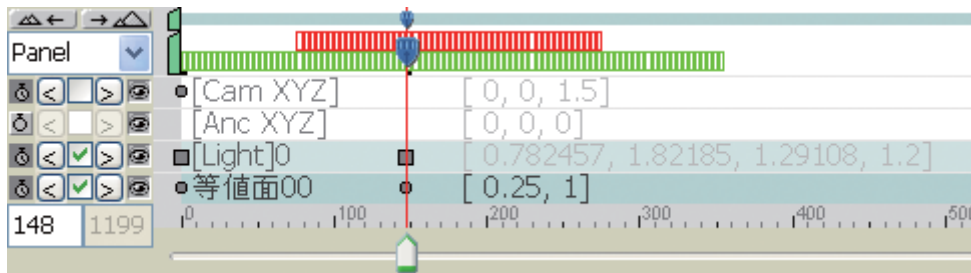


図 8. Oosawa で実装したタイムライン。Zindaiji と比較して編集できる対象は大幅に増している。

また、大解像度データを扱うにあたって、ネスティッドグリッド及びに八分木ツリーの形で構成されたデータから、カメラ位置に応じて必要なデータを取り出してレンダリングをする機能を実装している。八分木ツリーの場合は、図 9 のように

カメラに近い位置は子供の細かいデータを用いてレンダリングを行い、カメラから離れた場所では、親の大雑把なデータを用いたレンダリングを行なう。機能のテストとして行なったデータでは1つのボックスが256x256x256の解像度を持ち、深さ5までのツリー構造(8192x8192x8192相当)までレンダリングを行なうことが可能であった。残念ながら現時点では、これらの機能はレンダリング時のみに反映されて、プレビューには反映されない。

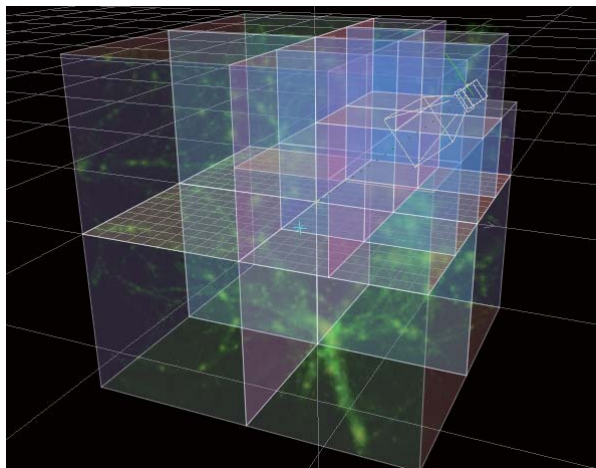


図9. 八分木ツリーで構成されたデータを、カメラ付近から再構築した場合に用いられるデータを表示した画面。カメラに近い場所では子供の細かいデータを用いてレンダリングを行なう。

また、ボリュームデータは物理現象のシミュレーションによって得られるようなデータ以外にも、生体の3次元構造など数多く存在し、連続的なスライス画像データとして扱われる場合も多い。そういったデータにも対応するために、連番画像ファイルの読み込みといった機能の実装も行なっている(図10)。



図10. 左) 連番画像ファイルの模式図. 中) それらを元に3次元表示した植物のタバコ培養細胞 (Bright Yellow No.2; BY-2) の小胞体. 右) カメラワークの自由な設定が可能のため、細胞の内部から見た視点なども得られる。(データ提供: 基礎生物学研究所, 真野昌二)

現在は、こうした高解像度データや時系列データを扱うための実装を行なった上に、さまざまな機能の追加を行なっている。データに対するレベル補正や、ガウス暈しといった簡単なフィルター処理、等値面や断面図の表示や、流線の表示、及びにそれらの形状をPovRay用のデータとして出力してレンダリングする機能などを実装している。今後も、可視化映像の作成に必要な機能などを順次付加してゆく予定である。

#### 4. まとめ

大規模なシミュレーションデータの可視化映像を作成するため、粒子系データ、ボリュームデータに対応したツール、ZindaijiとOosawaの開発を行なった。データに特化したツールの開発によって、通常は難しい大規模データにも対応した映像化をすることが可能になる。特に大規模なデータに対しては、構造化されたデータを取り扱う機能の実装が必要になる。

また時系列データからの映像製作には、タイムライン上でのメモリの管理機能とパラメータ編集の機能を実装することが重要である。

自家製ツールの限界の一つとして、レンダリング手法が限定されるためある程度以上のクオリティを持った画像、映像を得るのが難しいことがあげられる。この弱点は既存のレイトレーサーと連携するといった手法で補うことが可能である。

Oosawa の開発によって、Zindaiji と比較してより良い GUI の作成方法や八分木ツリーへの対応といったノウハウが得られたため、今後粒子系データの可視化ツールへの反映を行ないたい。

### 参考文献

- 1) 小久保英一郎ほか：4次元デジタル宇宙プロジェクト，情報処理，Vol. 45, No.12 (2004) pp. 1229-1233
- 2) 武田隆顕：大規模天文シミュレーションの可視化 - 国立天文台 4D2U プロジェクトでの試み，可視化情報学会誌 Vol. 28, No. 110 (2008) pp. 186-192
- 3) Takeda, T. and Ida, S.: Angular Momentum Transfer in the Protolunar Disk, *The Astrophysical Journal*, Vol. 560, (2001), pp. 514-533
- 4) Saitoh, T. R., Koda, J., Okamoto, T., Wada, K., Habe, A.: Tidal Disruption of Dark Matter Halos around Proto-Globular Clusters, *The Astrophysical Journal*, Vol. 640 (2006) pp. 22-30.
- 5) <http://4d2u.nao.ac.jp/html/program/mitaka/>
- 6) [http://4d2u.nao.ac.jp/src\\_4d2u\\_dome/src/Zindaiji/index.html](http://4d2u.nao.ac.jp/src_4d2u_dome/src/Zindaiji/index.html)
- 7) Crow F.C.: Shadow algorithms for computer graphics, *Proc. SIGGRAPH'77(1977)*, pp242-248.
- 8) L. Williams : Casting curved shadows on curved surfaces, *Proc. SIGGRAPH'78(1978)*, pp.270-274.
- 9) Persistence of Vision Pty. Ltd., Williamstown, Victoria, Australia. <http://www.povray.org/>
- 10) <http://th.nao.ac.jp/~takedatk/COMPUTER/OOSAWA/oosawa.html>