

大規模粒子シミュレーションデータ可視化ツール Zindaiji3 の開発について

武田 隆顕*

About new particle data visualization tool ” Zindaiji3”

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 once developed 32-bit application “Zindaiji” to visualize particle simulation data and used it to make high quality movies. Recently, simulations with much large particle number are commonly performed. Thus, we are now developing 64-bit application “Zindaiji3” for visualization of large particle number data. Here, we report the current status of “Zindaiji3” .

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も重要である. 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(Zindaiji)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きた. 近年より大規模な粒子シミュレーションが行われるようになり, そうしたデータに対応するべく新規に64-bit版として可視化ツール(Zindaiji3)の開発を開始した. Zindaiji3の開発の現状と今後について報告する.

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り, 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. 4D2Uプロジェクトでは主に後者に重きを置いた, 天文シミュレーションの可視化をおこなってきた.

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

* 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)

要な役割を果たすためである。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 (Zindaiji) の開発を行い、主にこのツールを用いて映像製作を行ってきた²⁾。Zindaiji は C++ で開発され、32-bit Windows 環境で動作するものとなっている。Zindaiji は、テクスチャの付加やカメラワークの編集機能の実装といった、CG 作成ソフトに近い機能を重視した粒子データ用の可視化ソフトウェアとして開発を行ったものである。Zindaiji は半径を持った球、もしくは光る点 (星) や簡易的なガス粒子を大量に表示する機能をもち、われわれはこれを用いて「月の形成」や「渦巻き銀河の形成」といった、粒子系の天文シミュレーションの映像化 (図 1) を行ってきた。

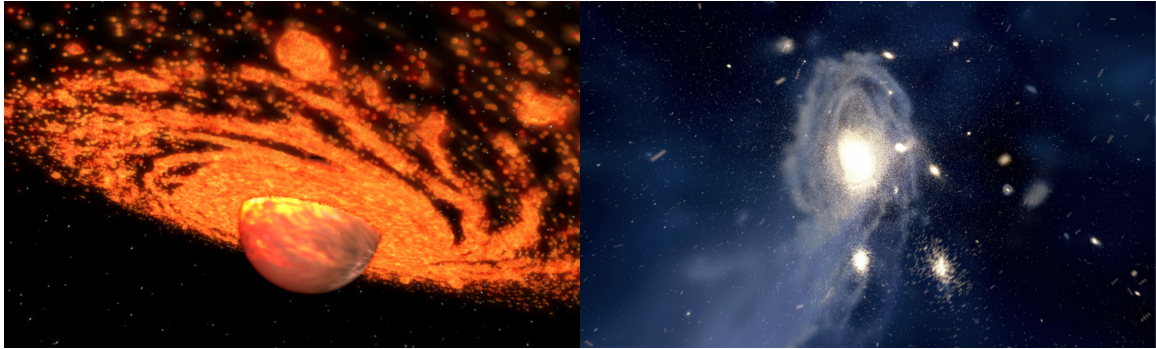


図 1. 左)「月の形成」のシーン。巨大衝突によって原始地球周りに撒き散らされた月の材料が集積する過程を示している³⁾。右)「渦巻銀河の形成」のシーン。星をあらわすスーパーパーティクルとガスをあらわす SPH 粒子を表示している。星とガスに 100 万粒子を用いたシミュレーションの可視化映像である。⁴⁾

初代 Zindaiji は 32-bit Windows アプリケーションとして開発されたため、主に利用できるメモリの制約から約 200 万粒子程度までが扱えるデータ量の上限となっていた。今日では、100 万粒子を用いる規模のシミュレーションは一般化し、より多くの粒子を用いた高分解能のシミュレーションも数多く行われている。

そこで、我々はより大規模なシミュレーションの可視化を行うために、64-bit アプリケーションとして Zindaiji3 の開発を開始した。Zindaiji3 では新たにコード開発を行い、大規模データに対応すると同時に、初代 Zindaiji の GUI における操作性上の不具合などを解消した、より強力で操作しやすいツールを目指して開発を行っている。セクション 2 では、Zindaiji3 の主な強化点について述べ、セクション 3 でまとめと今後について述べる。

2. Zindaiji3 における強化点

2.1 タイムライン GUI の強化

時間変化するデータを連続的に把握するためのもっとも素直な可視化方法は、時系列に沿ってスナップショット毎に連番画像を作成して、最後にそれをつなげた映像ファイルにすることである。しかし、本来は視点や時間、可視化用のパラメータなどを自由自在に操って、それをディスプレイ上に反映させることができることが望ましい。そうした、時間変化するデータを扱うためのアプリケーションで、もっともユーザーインターフェイスが進んでいるものとして映像編集用もしくは 3DCG アニメーション製作用のアプリケーションがあげられる。それらのソフトウェアでは、タイムライン上にカメラの位置やそのほか必要なパラメータのキーフレームを配置し、時間軸上でパラメータ値を自由に編集することで多種多様な映像の作成を行っている。初代 Zindaiji においてもタイムライン編集を軸にした可視化を行えるツールの開発を目指していたが、経験の不足から十分に目標を達成することができなかった。Zindaiji3 ではタイムライン編集機能を大幅に強化しパラメータのより自由な編集が可能となった (図 2)。

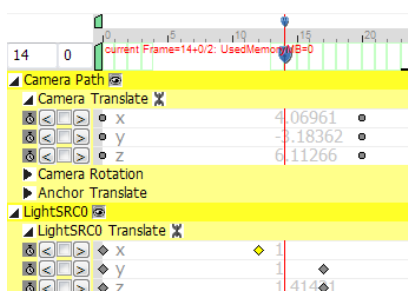


図2. Zindaiji3 になって強化されたタイムライン GUI. タイムライン上のキーフレームの配置やマウスによるドラッグといった機能や、表示非表示の選択を行える。タイムライン上部の緑のボックスは、メモリ上にデータが展開された時間ステップを表している。ユーザーの操作待ちの時間に、あらかじめ先の時間ステップのデータを読み込むプリフェッチ機能(セクション 2.3 参照)を実装することで、速い応答性と操作性の向上が図られている。

2.2 メモリ制限の軽減

32-bit アプリケーションでは、ソフトウェア上の制約から 2GB 程度のメモリしか使用することができないが、64-bit アプリケーションではメモリによる制限は大幅に軽減される。Zindaiji3 は 64-bit アプリケーションとして開発しているため、PC に積んでいる物理的なメモリ量、もしくは描画速度からくる制限が実質的に抜える粒子数の上限を決める。図 3 に、1 億体の粒子を用いたシミュレーションデータ内のフライスルー映像のスナップショットを示す。粒子の xyz 座標を単精度でメモリに保持していた場合、1 億体の粒子あたりに必要なメモリは 1.2GB になる。ただし、Zindaiji3 では一般的に時系列データを扱うため、速度、加速度や粒子の ID といった情報を保持している。そのため、それら全てのデータの保持に 6GB ほど使用する。また、元データとは別に補間済みのデータを作成しているため、それを保持するために 2GB メモリを使用しており、1 億体のデータを保持した状態で 8GB 程度のメモリ使用量となった。2012 年現在、16 ~ 32GB 程度のメモリを搭載した PC の調達是比较的容易であり、1 台の PC を用いて億の単位までは可視化を行うことができることになる。

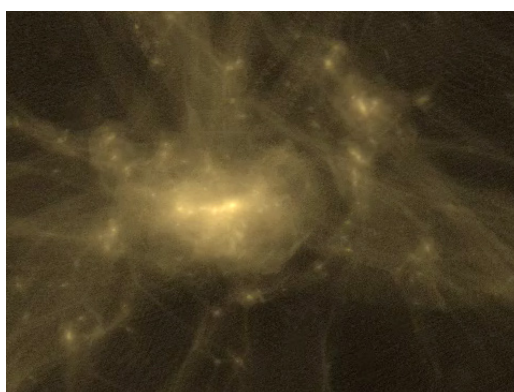


図3. 1億粒子を用いたシミュレーション内のフライスルーの一場面。(データ提供：石山智明) 5). 元のシミュレーションは約 170 億体の粒子を用いているが、そのうち特徴的な構造を含む領域から一億体分のデータを抜粋。映像の品質を上げるために、粒子は点 (GL_POINTS) ではなく、カメラの方向を向きテクスチャを張った小さな四角いポリゴンとして表示されている。

2.3 高速化と操作性の向上

64-bit 化に伴ってメモリに関する制約が少なくなったために、Zindaiji3 の実質的な有用性は、描画の速さや待ち時間の少なさに依存することになる。Zindaiji3 では実行速度の向上のためにアルゴリズムの見直しを行い、描画速度の向上とマルチスレッド化による待ち時間短縮が図られている。

OpenGL を用いた描画時間には、データの転送や描画そのものにかかる時間の他に、ステートの変更にかかる時間があげられる。ステートの変更とは、描画する色の変更や使用するテクスチャを変更する作業を指し、頻繁にステートの変更を行うと描画が遅くなることが知られている。ステートの変更をできるだけ少なくすることは 3DCG を用いたゲームなどで、高速化の基本的なテクニックとされている。

さまざまな物理量を持ち、色や粒子の種類（ガスと星など）が入り乱れたデータを描画する際には、最も単純に実装をすると図4の一番左の列のように、ループを用いて一つ一つの粒子の設定を行う描画ルーチンになる。

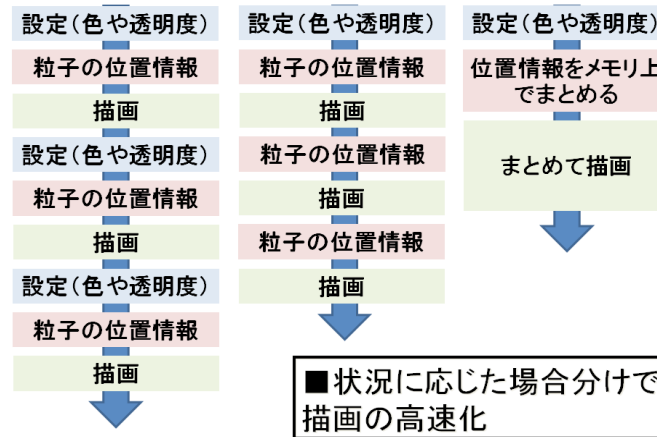


図4. Zindaiji3の描画に用いた場合分けを多用した効率化の模式図。左が最も単純なループを用いた描画ルーチン。

しかし、描画の設定によっては、こうした毎回のステータの変更は必要が無い場合がある。Zindaiji3においては、描画の設定に応じて場合分けを行い、ステータの変更が必要のないデータに対してはステータ変更の無いルーチンへ分岐して描画をすることで、描画速度の向上を図っている。また、連続して同一種類の粒子を描画するような場合には、ステータ変更をスキップすることで描画時間の短縮が図れる。ガスと星からなる銀河衝突のデータを用いて確認してみたところ、Zindaiji3では初代Zindaijiと比べてほぼ半分の時間で描画することができ、描画アルゴリズムの変更によって高速化を行えることが確認できた。図3で用いた1億体の粒子は、すべて同一の粒子として描画されており、ステータの変更はほとんど行われていない。このデータの場合では、グラフィックボードとしてNvidiaのGeForce 240を用いて、1枚の描画に約6秒の時間がかかっている。

上記の考えを拡張すると、ガスと星など描画方法の違う粒子が入り混じっている場合には、粒子の種類に応じてソートされた順番で描画をすることで描画速度の向上が期待できる。しかし、以下の理由により実際には適用できない場合が多い。SPHシミュレーションにおけるガス粒子のように不透明な粒子を描画する場合には、半透明なポリゴンを配置して表現を行う（図5）。この際、ポリゴンの相互の隠蔽の判定が行われるが、単純に粒子の配列順に表示を行うと、先に描画した手前のポリゴンの半透明部分によって奥のポリゴンが隠されるという現象が起きてしまう。そのため、半透明な物体の描画は奥からソートされた順番で行う必要がある。粒子の種類によってソートされた順番で描画を行うことができないことになる。そのため、現時点でZindaiji3では粒子の種類に応じたソートの実装はおこなっていない。

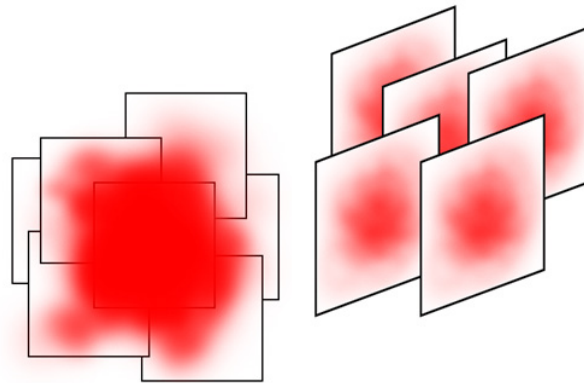


図5. SPH粒子などのガス粒子の表示方法の模式図。半透明な板を透明度に応じて奥から表示することで、ガスの分布を表現する。

動画を作成するには描画以外にも、さまざまな処理による待ち時間が存在する。Zindaiji3では、それらの処理をレッド並列で処理することで待ち時間の短縮を図っている。具体的にはデータ読み込みと補間、描画後の画像出力を描

画とは別スレッドで行えるように作成し、描画を行っている間、もしくはユーザーによる操作待ちの間に次の時間フレームの読み込みなどを行うことができるようになってきている (図6)。また、ユーザーの操作待ちの時間には、次の時間フレームのデータに限らず、その先の時間フレームも予め読み込むプリフェッチ機能を実装している。これらの実装によって、操作時及びレンダリング時の待ち時間が減少し、操作性は大きく向上している。

動画作成時の速度を主に律速するのは、データ読み込み処理と描画処理のうち、より時間のかかる処理になる。粒子をすべて光の点として表示する場合には、前述したように1億体の描画に約6秒かかっている。一方1億体分のデータファイルは6GB程度のサイズになるために、HDDからの読み込みには1分ほどかかる。この場合には、時間フレーム間を10コマほどで補間するアニメーションを製作すると、描画時間が1分ほどでデータ読み込み処理にかかる時間と同程度になり、マルチスレッド処理がもっとも効率的となる。

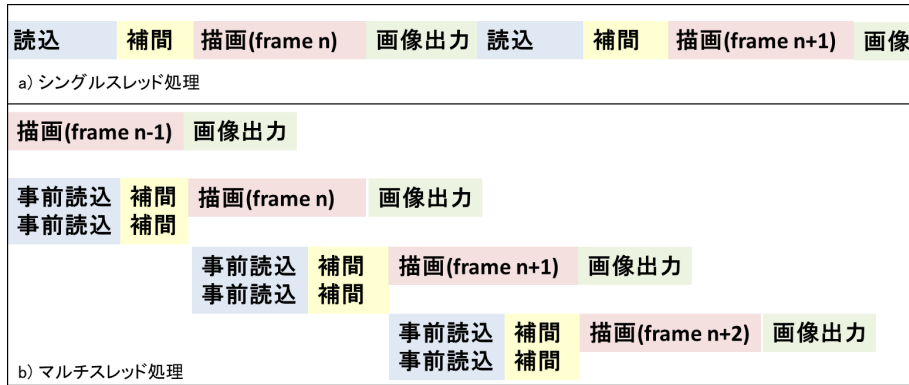


図6. Zindaiji3で実装したマルチスレッドによる並列化の模式図. a) シングルスレッドのみを使って、順番に処理を行う場合の模式図. b) 平行して処理を行うことで、高速化する場合の模式図. 事前読み込みと補間の処理が2行で書かれているのは、これらの処理をマルチスレッドで処理していることを表している。描画と画像出力はそれぞれ1スレッドを用いている。

2.4 描画品質の向上

大量の粒子データを描画するためには、描画品質の向上も重要である。たとえば単純に一粒子あたり一ピクセルで描画すると、大粒子数のデータでは簡単に画面を埋め尽くしてしまい、見たい構造が取り出せなくなる。また、図5のようなガス粒子を表示する場合、粒子数が多ければポリゴン一枚あたりの透明度が大きくなり、8ビット色深度のバッファ上で表示するには精度が足りなくなり、カラーバンディングが発生する。これらの問題から、Zindaiji3では16ビット色深度の2048×2048又は4096×4096ピクセルのバッファFrame Buffer Object (FBO)上に描画を行い、その結果を最後に画面上に表示をしている。これによりカラーバンディングの発生を抑制し、見かけ上の1ピクセルよりも小さいサイズの粒子の表示も扱うことができるようになった。

また、Zindaiji3は専門家以外の人に魅力的に感じる可視化映像を作ることが大きな柱になっている。そのため、追加機能としてより綺麗な出力をする機能を複数実装した。以下はそうした機能をいくつか列挙する。

大きさを持った球として粒子を表示する場合、影を実装することによって立体構造の表現力は大きく向上する。初代Zindaijiにおいても1つの光源からの影をシャドウボリューム法⁹⁾によって実装していたが、Zindaiji3ではFBO上で表示された画像を複数合成することによって、複数の光源からの影を表現する拡張を行った。光源を複数配置することで凹部に光が侵入しづらいという物理が再現され、複雑な形状でもその立体感が強調される (図7)。

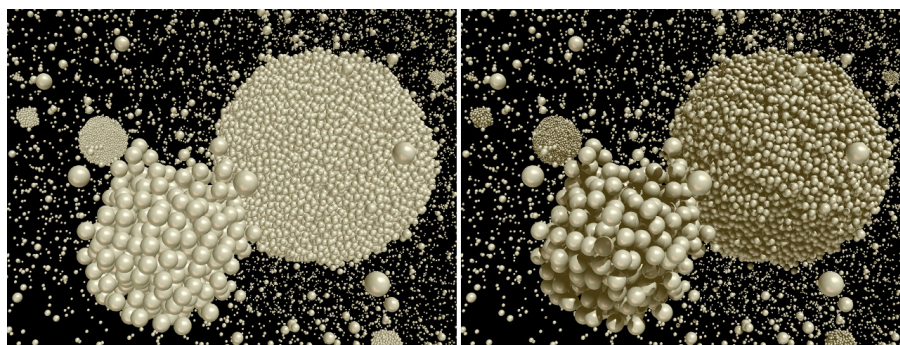


図7. 分子動力学計算による核生成過程のスナップショット. 左) 影のない表示. 右) 複数の光源からの影を入れた表示. 影によって凹凸がはっきりするほか、粒子の密集度に関しても受ける印象が変化する. (データ提供: 田中今日子)⁷⁾

コンピュータグラフィックにおいては、フォトリアルな表現を目指すほか、トゥーンレンダリングのように、より象徴度を高めたレンダリングを行うことがある。Zindaiji3 では、GLSL を利用することでトゥーンレンダリングの実装を行った (図8)。トゥーンレンダリングには、イラスト的でスタイリッシュな出力が得られるという単なる見かけの目新しさの他に、輪郭線が強調されることにより影に隠れた部分も認識しやすいなどの利点がある。

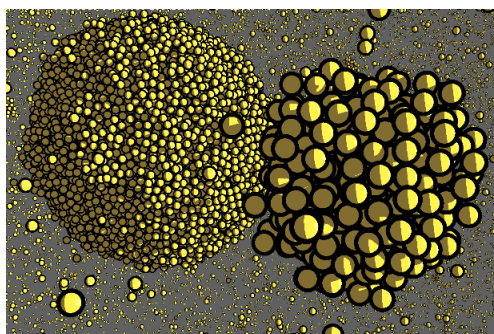


図8. トゥーンレンダリングを用いた表示. 用いたデータは図7と同じもの.

3. まとめ

Zindaiji3 によって、今まで用いていた初代 Zindaiji と比べ 100 倍程度、億の単位の粒子シミュレーションの可視化映像の作成を行うことができるようになった。演出を含めた映像制作においては、カメラワークや見た目の調整のためにトライアンドエラーを繰り返すことになり、できるだけ小回りの利く可視化プロセスであることが望ましい。そのため Zindaiji3 では、まず 1 台の PC でできる限り大規模なデータを綺麗に出力できる可視化ツールを目指し開発を行った。現時点では実装されていない機能はまだ多く残っているが、ある程度その目標を達成することができた。しかし、最新のスーパーコンピュータをフル稼働するようなシミュレーションでは、数百億といった粒子数を扱うことがある。そうした本当に大規模なデータの可視化は、もはや 1 台の PC での実行は難しく、少なくとも複数の PC を用いた作業が必要になると考えられる。次の大きな目標は、複数の PC を用いてより大規模な粒子データの可視化を行うことのできる、ネットワークレンダリングの実装となる。Zindaiji3 は、GUI 部分にマルチプラットフォームなライブラリである wxWidgets⁸⁾ を利用し、Windows/LINUX/Mac 上で実行することができる (ただし現在 Mac 版のみ 32-bit となっている)。現在ソースコードも含めて以下の URL で公開を行っているの、興味のある方はぜひダウンロードして試してみたい。

<http://www.cfca.nao.ac.jp/~takedatk/COMPUTER/ZINDAIJI3/Zindaiji3Top.html>

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., and Habe, A.: Tidal Disruption of Dark Matter Halos around Proto-Globular Clusters, *The Astrophysical Journal*, Vol. 640 (2006) pp. 22-30.
- 5) Ishiyama, T. (private communication)
- 6) Crow F.C.: Shadow algorithms for computer graphics, *Proc. SIGGRAPH' 77*(1977), pp242-248.
- 7) Tanaka, K., Tanaka, H., Yamamoto, T., and Kawamura, K.: Molecular dynamics simulations of nucleation from vapor to solid composed of Lennard-Jones molecules, *The Journal of Chemical Physics*, Vol. 134 (2011), 204313
- 8) <http://www.wxwidgets.org/>