

蓄積型シンプレクティック・レイトレーシングによる ブラックホールの可視化

佐藤 哲[†] 横矢 直和^{††} 竹村 治雄^{†††}

[†] 科学技術振興事業団 (CREST)

^{††} 奈良先端科学技術大学院大学 情報科学研究科

^{†††} 大阪大学 サイバーメディアセンター

シンプレクティック・レイトレーシングは、光線が複雑に曲進する状況下でコンピュータ・グラフィックスを生成する場合、数値計算の速度と精度の面から従来のレイトレーシング法よりも優れた性能を示す手法として提案された。シンプレクティック・レイトレーシングは様々な物理現象の可視化に使用できるが、本報告ではブラックホールを対象とする。ブラックホールは物理学上の事項であるため、直感的に理解しやすくすることによる、教育や研究の支援が目的となる。そのような目的に対しては、色々な異なる状況下での可視化結果や、動画として結果を閲覧できることが望ましい。しかしレイトレーシング法の概念を用いている以上、大量の画像を生成するには時間がかかる。そこで、データベースシステムと連携して数枚の画像間のマッチング情報を蓄積し、蓄積しておいたマッチング情報を用いて中間画像を次々と高速に生成するシステムを構築したので、ここに報告する。

Visualization of Black Holes by an Accumulative Symplectic Ray Tracing

Tetsu Satoh[†] Naokazu Yokoya^{††} Haruo Takemura^{†††}

[†] Japan Science and Technology Corporation (CREST)

^{††} Graduate School of Information Science, Nara Institute of Science and Technology

^{†††} Cybermedia Center, Osaka University

This paper reports a method of generating movies from images rendered by a symplectic ray tracing with database system. The symplectic ray tracing has been proposed as a method for generating computer graphics by curved light rays. By applying the symplectic ray tracing, the rendering performance is improved in calculation time and accuracy. The symplectic ray tracing is applicable to visualization of various phenomena. However, this report focuses on visualization of black hole spacetime. The black hole is a topic in physics so that the present purpose of visualizing the black hole is for support of education and research. For this purpose, it is required to visualize results under various conditions and to generate movies. However, it is time consuming to generate a great number of images as far as the concept of ray tracing is used. To solve this problem, we have developed the following method. First, a number of calculated images are accumulated into database system. We then obtain interpolated images efficiently.

1 はじめに

光線が直進しない状況下での物理現象を可視化するために、非線形光線追跡法 [1] と呼ばれるコンピュータ・グラフィックス (CG) 作成手法が用いられている。光線が直進しない状況とは、例えば固体や気体の密度の変化により光線の屈折率が変化する媒介の中を光線が進行する場合 [2]、空間がゆがむ事により、空間に沿って進む光線も曲進してしまう場合 [3] などがある。このような場合、光線の軌道は微分方程式で表されることが多く、効果的に数値計算をすることが課題となる。本報告では、可視化対象とする物理現象として、宇宙空間にブラックホールが存在してその影響により光線が曲進する状況を想定する。

ブラックホールを可視化した結果として得られる画像の一般的な問題点は、可視化結果の解釈が難しいことである。例えば図 1 は、画像の中心近くの 3 つの円柱の交差点の黒い円の部分に球状のブラックホールがあり、背景の銀河の画像や平面がゆがんでいる状況の可視化例である。ブラックホールが存在しない場合は図 2 のようになる。つまり、図 2 のような宇宙空間にブラックホールを配置すると図 1 に観測されることを示しているのであるが、非専門家が見ても、なぜ平面の一边だけが突然曲がるのか、背景の画像が円状にゆがんでいるのはどうしてなのか等、すぐに理解することは難しい。これらの現象は、光線が曲進する軌道を考えれば理解できることであるが、可視化する以上、直感的に理解できる方が望ましい。直感的な理解を助ける方法としては、静止画ではなく動画を作成することが考えられる。上記の例でも、ブラックホールの強さがゼロから徐々に強くなっていく様子を見ることができたり、視点を移動させて違う角度から見るのが可能であれば、少しずつ変化していく様子からどのようにブラックホールの効果が現れてくるかが直感的に理解できるであろう。しかし、光線が直進する状況で光線追跡法を用いて動画を提示することが実用的になったのは比較的最近であり、光線が直進しない状況でのリアルタイム・レンダリングは難しい。

そこで本報告では、異なる状況でのブラックホール時空の可視化結果の 2 画像で、両方

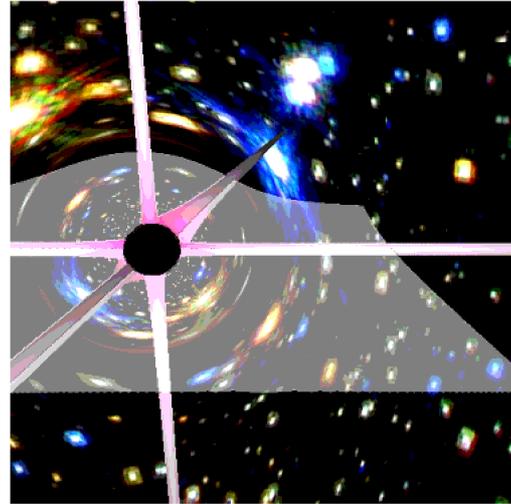


図 1: ブラックホールの可視化例

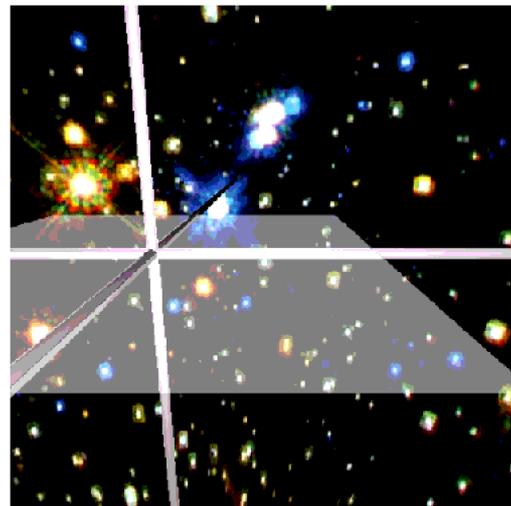


図 2: ブラックホールが存在しない場合

の画像に対応する画素があれば、どの画素とどの画素が対応するかが分かることを利用して、2 画像の間の画像を補間生成する手法について述べる。画素の対応が分かる理由は、可視化対象を非常に単純にモデル化したからであり、新しく提案する概念ではない。従って複雑なモデルに対する本手法の適用については今後の課題となる。

2 シンプレクティック・レイトレーシング

シンプレクティック・レイトレーシング [4] は、非線形光線追跡法の新しいアプローチとして提案された。その特徴は、従来の非線形光線追跡法で使用されていたように光線の座標

だけを数値的に計算していくのではなく、光線の座標の変化度に対応する運動量も同時に計算する、ハミルトンの正準方程式に基づき光線の座標を計算する点である。ハミルトンの正準方程式は、ハミルトニアンと呼ばれる一つのスカラー関数より任意の次元の物体の運動方程式を導くハミルトン力学の基礎となる方程式である。従来手法より制限される点としては、基礎とする方程式の性質上、ハミルトン力学で記述できない現象の可視化には適用できない点があげられる。光線の軌道を計算する方程式のみが分かっている場合にシンプレクティック・レイトレーシングを適用しようとする、その方程式を導くことができるハミルトニアンを求めなければならないという逆問題が発生してしまう。しかしハミルトニアンが求まった場合には、精密に高速な光線追跡が可能であることが報告されている [5]。

より高速とは言え、非線形光線追跡法を基礎にした場合は実時間での描画は難しいことが知られている [6]。そこで今回試作したものが、次節で述べるデータベースとの連携システムである。

3 データ蓄積型シンプレクティック・レイトレーシング

3.1 データ蓄積型シンプレクティック・レイトレーシングの概略

本研究の目的は、異なる状況での可視化結果を連続的に提示することである。異なる状況とは、ブラックホールの大きさ、観測者の位置や視線方向が異なることを意味する。なぜ連続的に提示するのが難しいかという一画像を生成するのに時間がかかるからであり、時間がかかるのは光線の追跡計算の部分である。従って、一度計算した光線の軌道を蓄積しておけば、その計算結果を用いた画像の提示は実時間で可能である。複数画像を作成した時の光線の軌道を蓄積しておけば、画像間の変化が不連続でない場合は補間により中間画像を生成することができる。中間画像生成は光線を追跡しなおすわけではないので、高速に生成可能である。

従来の非線形光線追跡法は光線を追跡しては描画するという処理を繰り返していたが、本

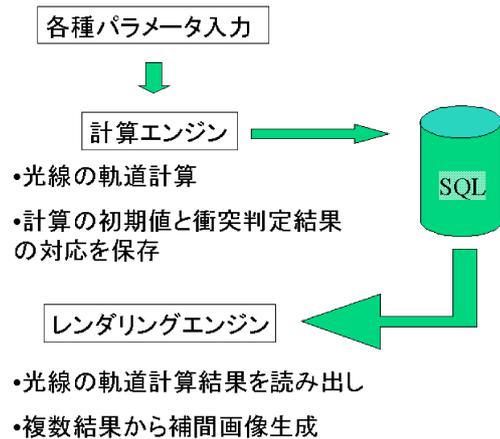


図 3: データの流れ

手法では計算処理と描画処理の二つに分離する。計算処理エンジンは、光線追跡と追跡結果をデータベースに蓄積する処理の二つを担当する。描画処理エンジンは、データベースから蓄積されている計算結果を呼び出すことと、呼び出した結果から補間画像を作成する二つの処理を担当する (図 3 参照)。計算エンジンの部分では、光線の軌道の並列計算と並列データベースアクセスが主要な処理となる。レンダリングエンジンの部分では、補間画像生成の並列計算が主要な処理である。システムを通じて、並列計算には OpenMP ライブラリを、データベースアクセスには PostgreSQL ライブラリを用いている。様々な処理系で動作する上記ライブラリを用いることで、プログラムの移植性が高くなっている。

3.2 計算エンジンの処理の流れ

計算エンジンは、次の流れに基づいて光線の軌道を計算し、結果をデータベースに蓄積する。

- (1) ブラックホールや観測者の位置、観測方向、ブラックホールの大きさなどのパラメータを決定する。
- (2) 観測者の視野平面内の各点 (i, j) について、以下を繰り返す。
 - (a) 観測者の位置から (i, j) へ向かうベクトルを視線ベクトルとする。
 - (b) 観測者の位置を光線の初期位置、視線ベクトルを光線の初期方向ベクトルとする。

ルとし、光線が配置オブジェクトと交差するまで以下を繰り返す。

- (i) 光線の現在の位置 x_k から、シンプレクティック数値解析により次の位置 x_{k+1} を求める
- (ii) 線分 $x_k x_{k+1}$ により光線を近似し、配置オブジェクトと交差判定をする
- (iii) 交差していた場合、光線の初期位置、初期方向ベクトルとともに交差点の座標、交差したオブジェクト名などを記録する。交差していなかった場合、(i)へ。

3.3 レンダリングエンジンの処理の流れ

レンダリングエンジンは、次の流れに従って蓄積データから画像データを生成する。

- (1) 2枚の画像のデータをデータベースから読み込む。一枚目を(A)、二枚目を(B)とする。
- (2) 補間する画像の数を N 枚と決める。
- (3) (A) のスクリーン上の各点 (i, j) について、以下を繰り返す。
 - (a) 観測者の位置から (i, j) へ向かうベクトルを視線ベクトルとする
 - (b) A のデータベースより、視線ベクトル方向に発射した光線がオブジェクトと交差した座標を読み出す。
 - (c) B のデータベースより、A におけるオブジェクトとの交差座標と同じ交差座標を持つBの視線ベクトルに対応するBのスクリーン上の点 (i', j') を読み出す。
 - (d) (i, j) と (i', j') が対応点となるので、 (i, j) と (i', j') を結ぶ直線を N 個に区切り、連続的に画像が変化すると仮定して N 個の画像に対応点の色をプロットする。

以上の処理で N 枚の補間画像が得られるが、ブラックホールの可視化を対象とする場合、同じ映像が入れ子状に複数回提示される場合がある。つまり、A の (i, j) に対応する B

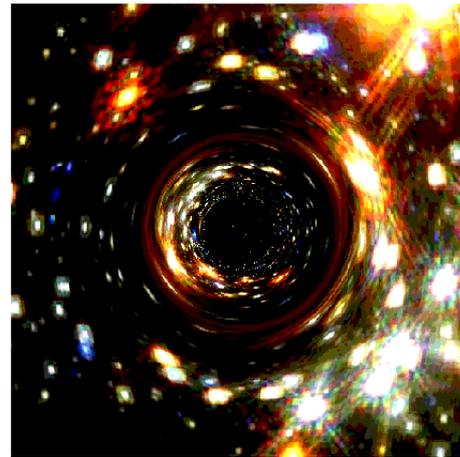


図 4: 動画の開始画像

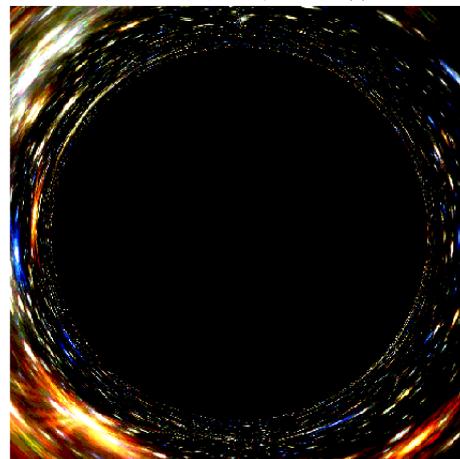


図 5: 動画の終了画像

の点が複数検索される可能性がある。その場合、光線の軌跡の長さなどから対応点 (i', j') を判断しなければならない。

4 レンダリング時間計測結果例

実装には、計算機は SGI 社の Onyx2 InfiniteReality (R10000, 195MHz, メインメモリ 3Gbytes) 用い、各計算処理、データベース処理を同機上で行った。

図 4, 図 5 のような 2 枚の異なる状況のブラックホールの可視化結果を連続的な動画で示すことを考える。これは、ブラックホールの大きさを連続的に変化させた場合と考えられる。

一般に、ブラックホール時空を可視化するための計算時間は、ブラックホールの大きさや位置などの状況によって、速い場合は 5 分

であっても遅い場合は数百時間かかるほど激しく変化する。今回の実装では、従来のシンプレクティック・レイトレーシングでは図4、図5を200×200画素の大きさでレンダリングした場合、5分強の時間がかかった。つまり、図4から図5に変化する様子を秒間3枚、10秒間で30枚の変化の遅い動画で提示すると、5分×30=150分の時間がかかる。蓄積型シンプレクティック・レイトレーシングの場合、計算よりもデータベースへのアクセス時間がネックとなる。これはPostgreSQLのトランザクション機能を使用しているため、トランザクション処理に時間がかかるためである。上記の従来のシンプレクティック・レイトレーシングを用いた場合と全く同じ状況で蓄積型シンプレクティック・レイトレーシングを実行した場合、計算エンジン、つまり光線の軌道計算及び計算結果のデータベース構築には44分弱の時間がかかった。レンダリングエンジン、つまりデータベースからデータを読み出して画像を生成する場合は一画像で4分弱であった。従って、計算エンジン部分の2画像の計算とデータベース作成に44分×2=88分、レンダリングエンジン部分の2画像分のデータ読み込みに4分×2=8分の時間がかかる。補間画像生成は30枚程度だと1分以内なので、測定誤差とみなしても良いくらい短い。

上記のように、状況の変化が些細な場合でも全て計算しなおす通常のシンプレクティック・レイトレーシングでは、30画像を生成するのに確実に150分の時間が必要となる。蓄積型シンプレクティック・レイトレーシングの場合は96分の時間が必要であった。しかし蓄積型の場合は、88分かけて2画像分のデータを生成しておけば、次からはデータ読み出し時間の8分プラス若干の補間画像生成時間で動画が生成できる。例えば60画像により動画を生成する場合、通常のシンプレクティック・レイトレーシングでは300分の時間が必要となるが、蓄積型シンプレクティック・レイトレーシングでは補間画像生成時間は非常に短いのでやはり約96分で済み、計算エンジンの計算時間を前処理として考慮外とすれば約8分となり、通常のシンプレクティック・レイトレーシングとの差は数10倍にもなる。蓄積型シンプレクティック・レイトレーシングの短

所としては、補間画像はやはり近似画像と考えられるため、理論に基づく正確さの面では通常のシンプレクティック・レイトレーシングには劣る可能性がある。また、開始画像以降に新たに画像中に入ってきた画素がある場合、補間が不可能となるために画像サイズを小さくする必要がある。つまり、開始画像と終了画像とで対応が取れない画素については補間画像が生成できない。この問題を回避するには、あらかじめ大き目の画像データを蓄積しておく必要が生じ、計算時間が長くなる。図4から図5のように変化する場合は、画素はむしろ画像外へ移動していくため、新たな画素が入ってくることはなくこのような問題は生じない。

実験結果を考察すると、長い時間を消費してまでトランザクション機能完備のデータベースを使用する必要があるかどうかという疑問が生じる。トランザクション機能の弱いフリーソフト（市販されていず、無償で入手可能なソフトウェア）のデータベースとして有名なものではMySQLやBerkeleyDBなどがあり、PostgreSQLよりアクセスは速かった。また、データベースシステムを使用せずに計算データを直接ファイルに出力した場合も、例えばPostgreSQLを用いて1時間かかった場合は10分程度で終了した。しかし、二つの理由から本研究ではPostgreSQLを採用した。まず第一に、トランザクションによるロバストなデータアクセスが必須だったからである。単純なファイル書き込みは勿論高速だが、シングルCPUによる計算と書き込みの実験でも、OSレベルでの書き込みと実際にハードディスクへ書き込みが終了するまでの時間にタイムラグがあり、同期がとれずにファイルの内容が確実に壊れた。そもそも単純なファイル書き込みと読み込みで十分なパフォーマンスが得られるのなら、データベースシステム自体が不必要となる。また、本研究では並列・分散処理による高速化が前提条件となるため、データの一貫性が保証されないデータベースシステムだとアクセスが激しくなるに連れパフォーマンスの低下が見られる。第二に、データベースシステムは蓄積されたデータの保守性が優れている。蓄積されたデータの内容を容易にチェック可能なことは勿論、SQLの命

令を用いて特に詳細に検討したい一部のデータを抽出したり，別な計算機にデータベースを移動させたりネットワークを通してデータベースにアクセスしたり，Webと連携させたインターフェースを容易に作成可能であるなどの良い特徴がある．従って，対象がブラックホールという直感的な理解が難しい問題であるために，可視化システムの作成の際に数値データを詳細に検討する必要性が生じる可能性があるため，データベースの操作が容易なPostgreSQLを採用した．単純に計算結果をファイル出力しただけでは，大量の計算結果の中から必要なデータを取り出すことは非常に難しい．

5 おわりに

本報告では，データベースシステムと連携させたデータ蓄積型シンプレクティック・レイトレーシングの試作例について述べた．そしてレンダリング実験により，一定の仮定の元では忠実にシンプレクティック・レイトレーシングを適用するよりも数10倍以上の速度で画像を生成できることを報告した．ただし，データベースシステムへのアクセスが処理速度低下のボトルネックとなること，補間画像生成のアルゴリズムを改良する必要があることなど，今後解決しなければならない問題点も確認された．これらの問題を解決するために，現在完全自動拘束条件無し画像マッチングシステム[7]の採用実験を行っており，現在の蓄積型シンプレクティック・レイトレーシングの実装と融合させることで改善を図りたいと考えている．

謝辞

本研究の一部は，科学技術振興事業団(JST)の戦略的基礎研究推進事業(CREST)「高度メディア社会の生活情報技術」プログラムによる．また，有益な助言を与えてくれた奈良先端科学技術大学院大学マルチメディア統合システム講座の前田亮氏と鈴木優氏に感謝する．

参考文献

[1] E. Gröller. Nonlinear ray tracing: Visualizing strange worlds. *The Visual Computer*, Vol. 11, pp. 263–274, 1995.

- [2] 斎藤泰，牧野光則，大石進一．レイトレーシング法を用いた異方性不均質透明体の表現．信学論，Vol. J76-D-II, No. 8, pp. 1755–1762, 1993.
- [3] 山下義行．相対性理論のコンピュータグラフィックス．日本物理学会誌，Vol. 53, No. 11, pp. 819–825, 1998.
- [4] 佐藤哲，岩佐英彦，竹村治雄，横矢直和．シンプレクティック・レイトレーシング：ブラックホール時空での光線追跡．情処論，Vol. 42, No. 3, pp. 456–464, 2001.
- [5] 佐藤哲，岩佐英彦，竹村治雄，横矢直和．高速シンプレクティック・レイトレーシング：入れ子宇宙の可視化．情処論，Vol. 42, No. 10, 2001 (採録決定)．
- [6] A. Hanson and D. Weiskopf. *Visualizing Relativity*. SIGGRAPH2001 Course notes, 2001.
- [7] Y. Shinagawa and T. L. Kunii. Unconstrained automatic image matching using multiresolutional critical-point filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20, No. 9, pp. 994–1010, 1998.