

宇宙科学データの可視化 —モバイル環境への応用について—

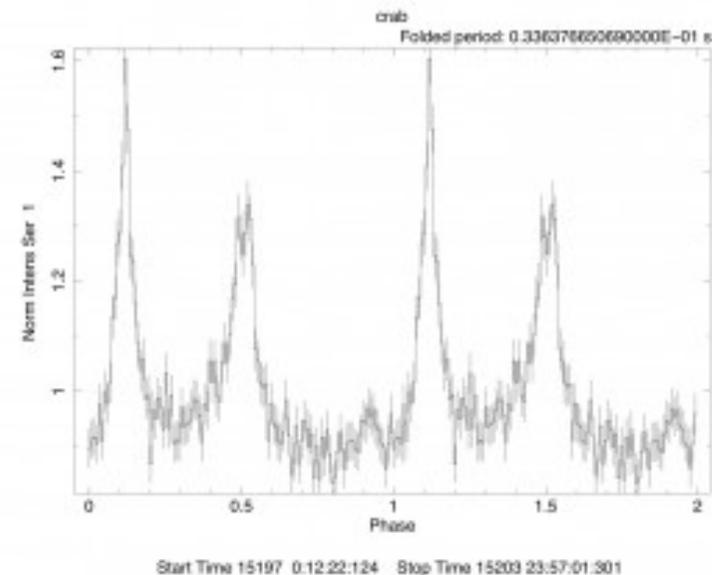
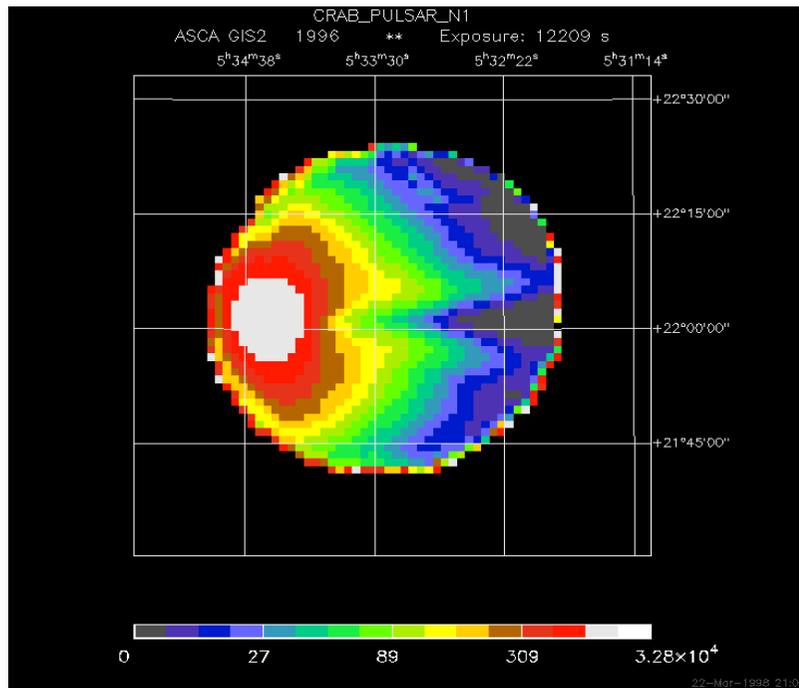
三浦 昭

宇宙航空研究開発機構

宇宙科学研究所

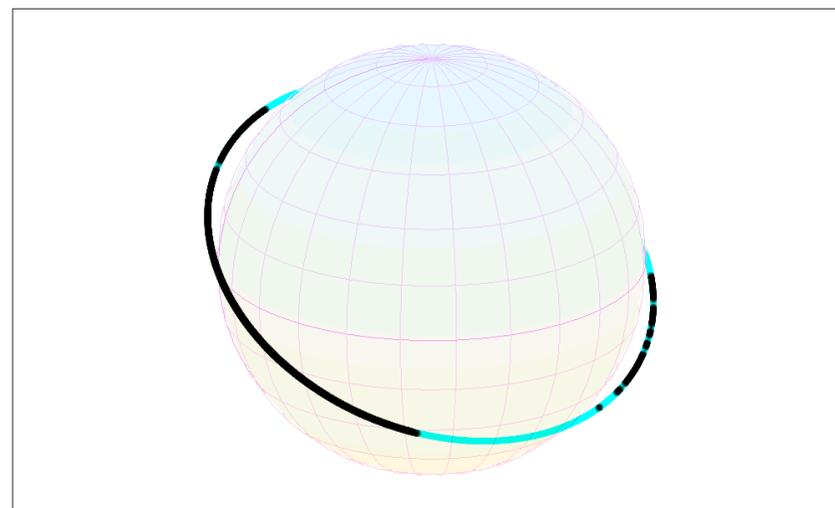
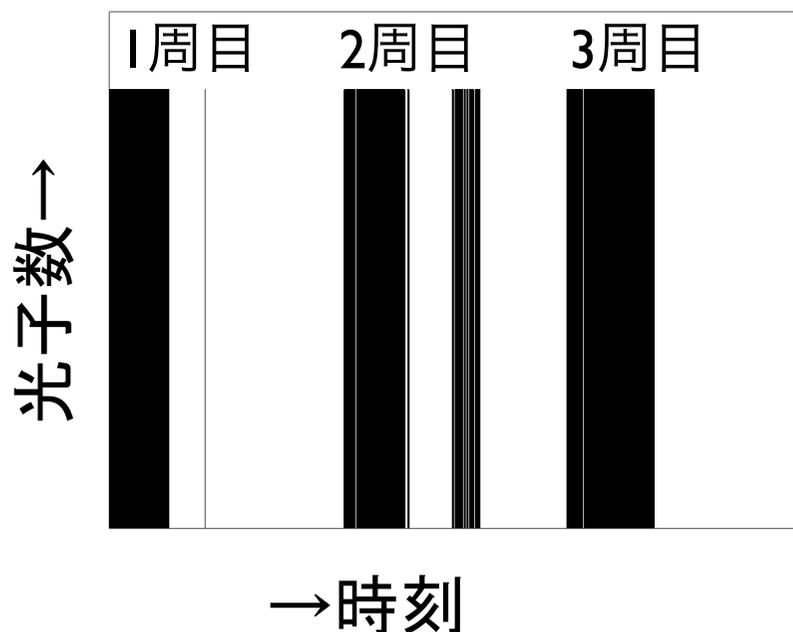
今日の素材: かにパルサー

- ・ 毎秒30回転している中性子星
- ・ 左: ASCAがとらえたかにパルサーのX線画像
- ・ 右: 全天X線観測装置 MAXI がとらえた周期変動



ASCAの観測データ(かにパルサーの例)

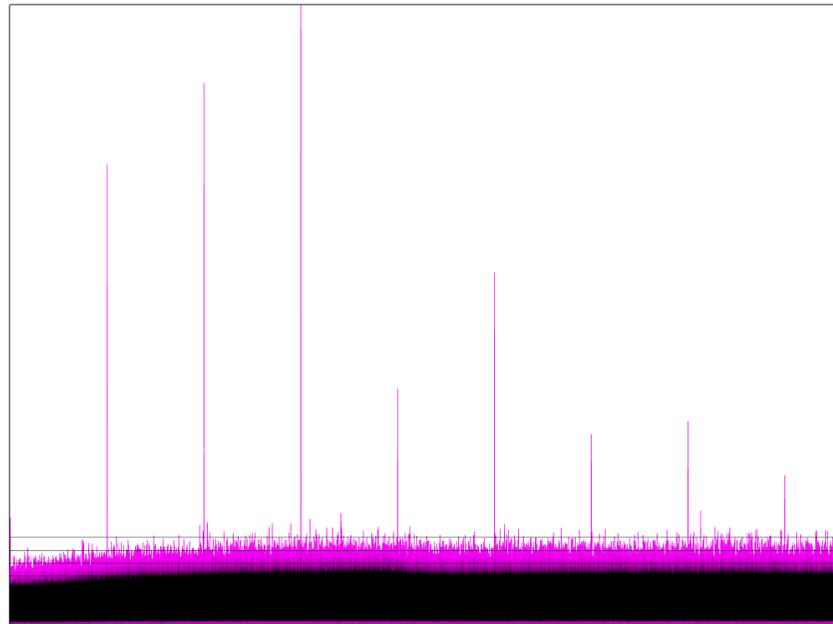
- ・ 時系列データである
 - 画像に合成するだけでなく、時刻に沿って見ることも可能
 - ・ 観測データには、時刻情報が記録されている
 - 左: 観測データ
 - 右: 観測対象(かにパルサー)から見た「あすか」の軌道
 - ・ 黒色: 観測データの取得範囲、水色: その他の軌道



パルサーの周期を検出

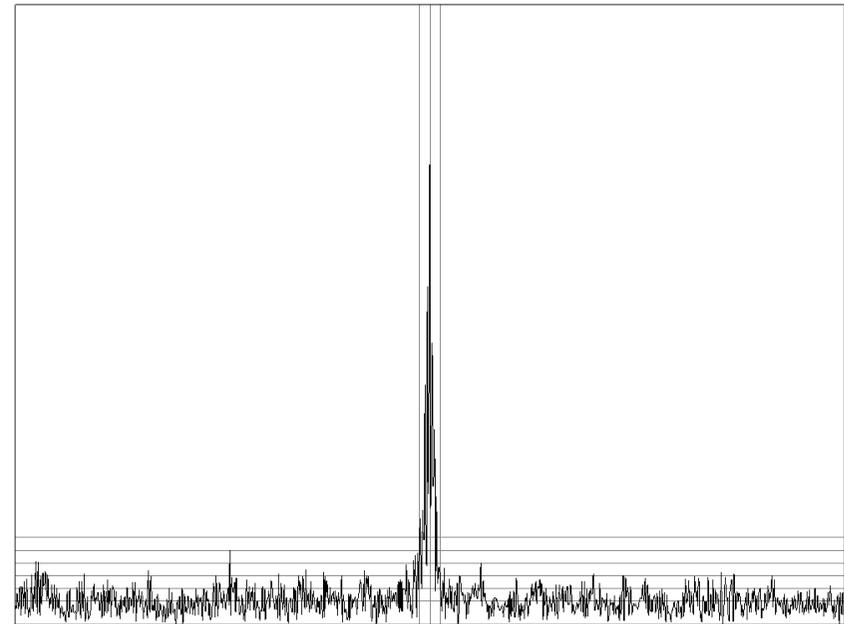
- ・ FFTでパルス周期を検出

↑強度



→周波数

大域

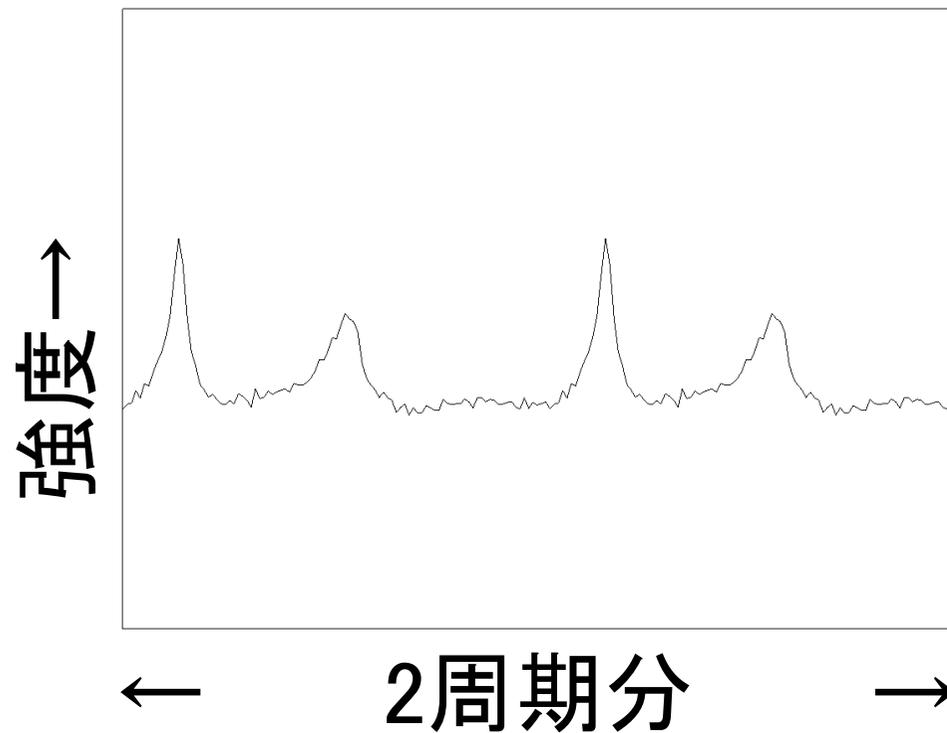


→周波数

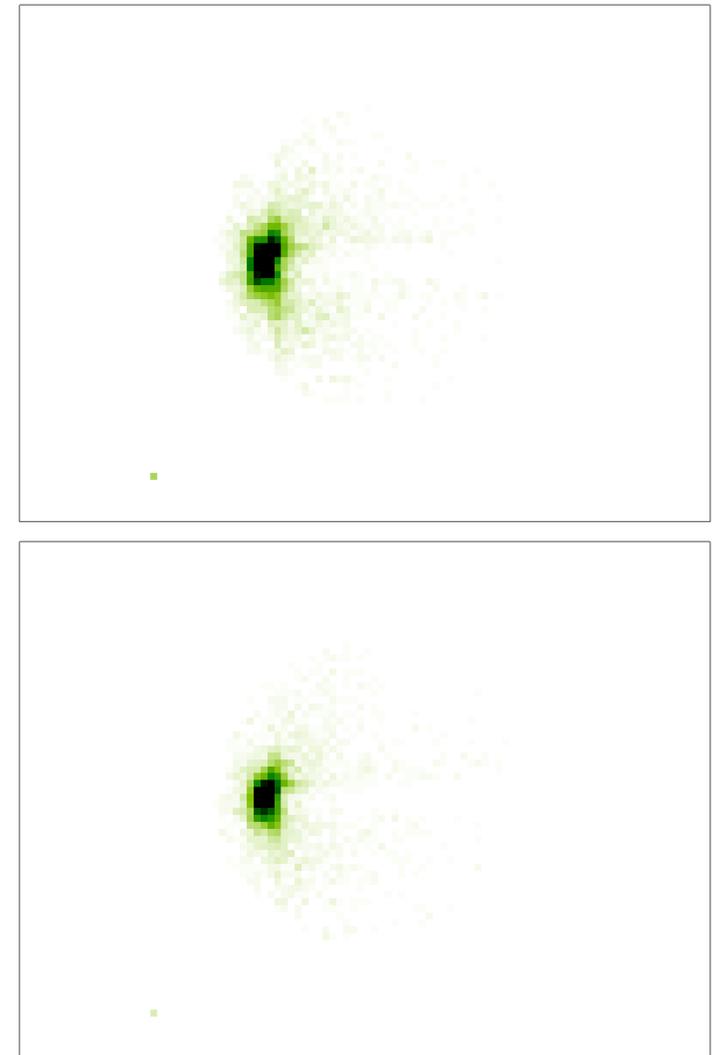
パルス周期付近の拡大

パルスの輝度変化

- 観測データをパルス周期で折り畳む
 - 輝度変化が再現できる



検出された周期で折り畳む
(周期を微調整)



輝度変化を動画に

動画で表現



モバイル環境を考慮

- **環境条件**
 - CPU、グラフィックス
 - かなり速くなった
 - ネットワーク環境
 - 相対的に遅いとは言え、ADSLとは互角？
 - 数Mbps～
- **データの提供方法1: 予め映像化して配信**
 - すぐにでも始められる
 - e.g., Youtube
- **データの提供方法2: 科学データをダウンロード**
 - 端末側で一昔前のデスクトップ機並みの処理は可能

宇宙科学映像の配信

- **映像配信のメリット**
 - 待たずにデータを視聴できる
 - バッファリングの時間は少々必要だが、待ち時間は少ない
 - 端末側でデータを保持する必要が無い(キャッシュ程度)
- **映像配信のデメリット**
 - 融通が利かない
 - 事前に映像化
 - パラメータを固定した映像になる
 - もしくは想定されるパラメータの組み合わせ分の映像を用意
 - 再生にかかるトータルの通信量は意外に大きい

宇宙科学データのダウンロード

- **科学データをダウンロードするメリット**
 - 端末上でデータを操作できる
 - 一度ダウンロードしたデータは、いろいろな切り口で再生可能
 - 通信量の低減
 - パケット定額制とは言え、極端な通信量に対しては制限がかかる
- **科学データをダウンロードするデメリット**
 - ダウンロードの待ち時間が長くなる
 - 一旦全データを読み込んでからデータを操作
 - データを保持するためのメモリが必要
 - FITSデータを解釈するメモリと可視化処理のためのメモリが必要
 - CPUパワーを要する

例えばこんな環境

- **科学データをダウンロードするメリット**
 - 端末上でデータを操作できる
 - 一度ダウンロードしたデータは、いろいろな切り口で再生可能
 - 通信量の低減
 - パケット定額制とは言え、極端な通信量に対しては制限がかかる
- **映像配信のメリット**
 - 待たずにデータを視聴できる
 - バッファリングの時間は少々必要だが、すぐに再生できる
 - 端末側でデータを保持する必要が無い(キャッシュ程度)
- **双方のメリットを享受できるか？**

端末側の環境(ノートPC)

- **高性能CPU**
 - 性能/クロック比の向上
 - 2コア、4コア、...ノートPCでも
- **大容量RAM**
 - 4GB~8GB
- **グラフィックス(GPU)**
 - HD~Full HD
 - モデルによっては、GPGPUも
- **高速ネットワーク**
 - 家庭にもギガビットインタフェース
 - 無線LANも100Mbps超
 - モバイルPCは数Mbps~数十Mbps?

最近のスマートフォン(Xiの例)

- **高性能CPU**
 - >1GHz
 - デュアルコア
- **大容量RAM**
 - 1GB
- **グラフィックス(GPU)**
 - 1280 x 720
 - Open GL ES
- **高速ネットワーク**
 - 最大75Mbps(屋内の場合)
 - 実効の速度はもっと遅い

(参考)我が家のパソコン

- **高性能CPU**
 - 2GHz
 - シングルコア
- **大容量RAM**
 - 768MB
- **グラフィックス (GPU)**
 - 1280 x 1024
 - Open GL
- **高速ネットワーク**
 - ~28Mbps
- **購入当時は、速かったのですが...**

低速ネットワーク対策(過去の事例)

- **静止画像の表示**

- **プログレッシブJPEG、インタレースGIF**

- 14.4kbps, 28.8kbps, etc.でも画像の概要を早く表示

- **IMGタグの属性**

- `lowsrc`, `alt`

- **動画配信**

- **プログレッシブダウンロード(疑似ストリーミング)**

- ダウンロード完了前に再生可能

- ファストスタート(QuickTime)

- 再生に必要なパラメータ類をファイルの先頭に配置

- 最近有名なのはFLV

- **ストリーミング(RTSP)**

- ネットワーク環境に応じて複数のビットレートを用意

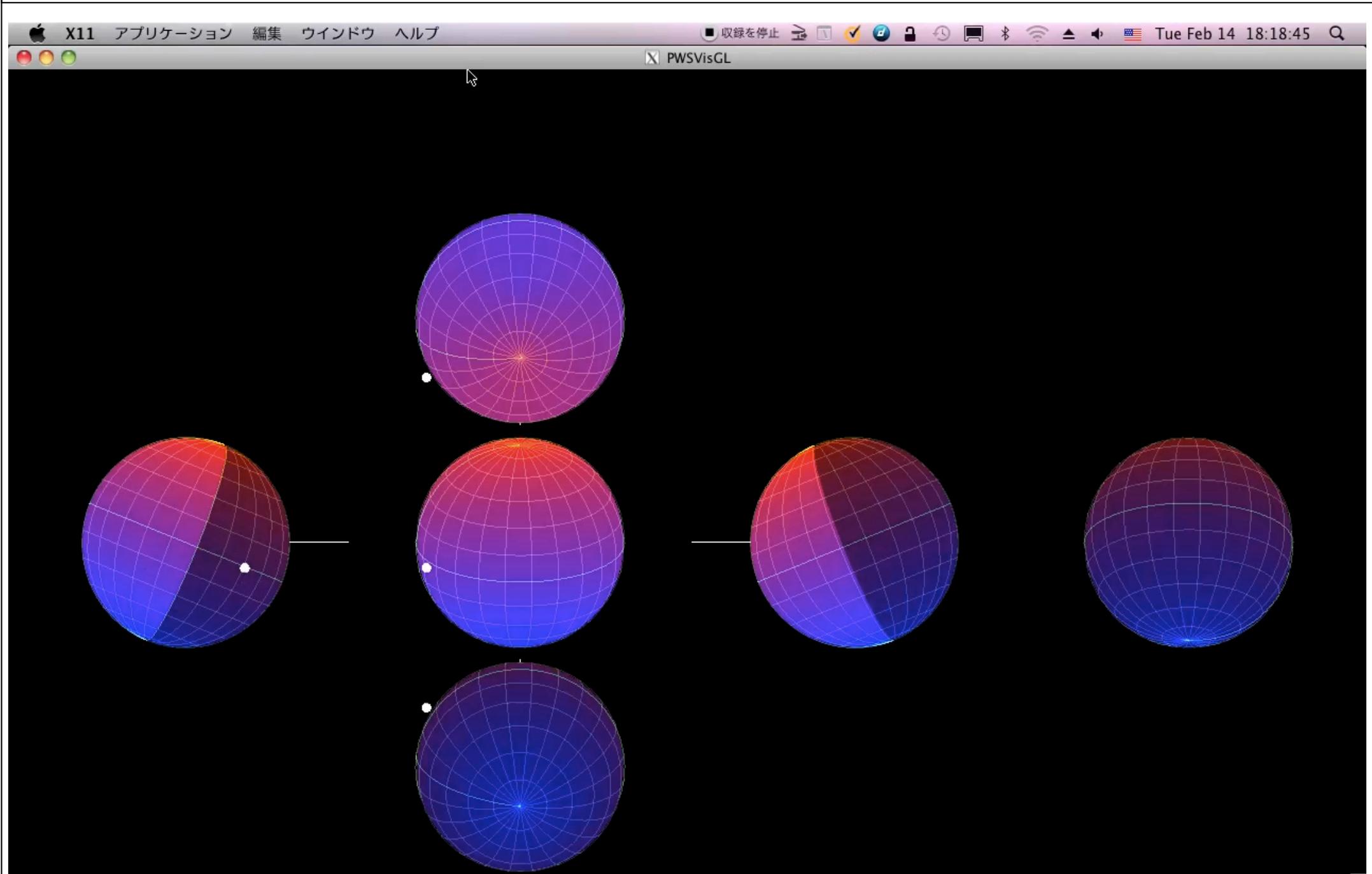
宇宙科学データは“プログレッシブ”対応か

- ダウンロードしながら可視化できるデータとは
- FITS形式
 - ヘッダ-(データ-)ヘッダ-データの形式で格納
 - 可視化に必要なパラメータは、ファイルの先頭にある
 - (衛星の位置情報等、別ファイルになったパラメータも有り)
- X線天文衛星の観測データ
 - 1つ1つの光子を検出した時刻、位置、エネルギー等を記録
 - 時系列データ(イベントが時刻順に並んでいる)
 - データの一部を切り出しても、X-Y方向の画像を再現できる
 - 画像は粗くなるが、時間変化も表現できる
 - 一種の動画データ
 - c.f. 天体写真はX-Y方向に走査したデータなので、データの一部を切り出したら、画像の一部しか再現できない

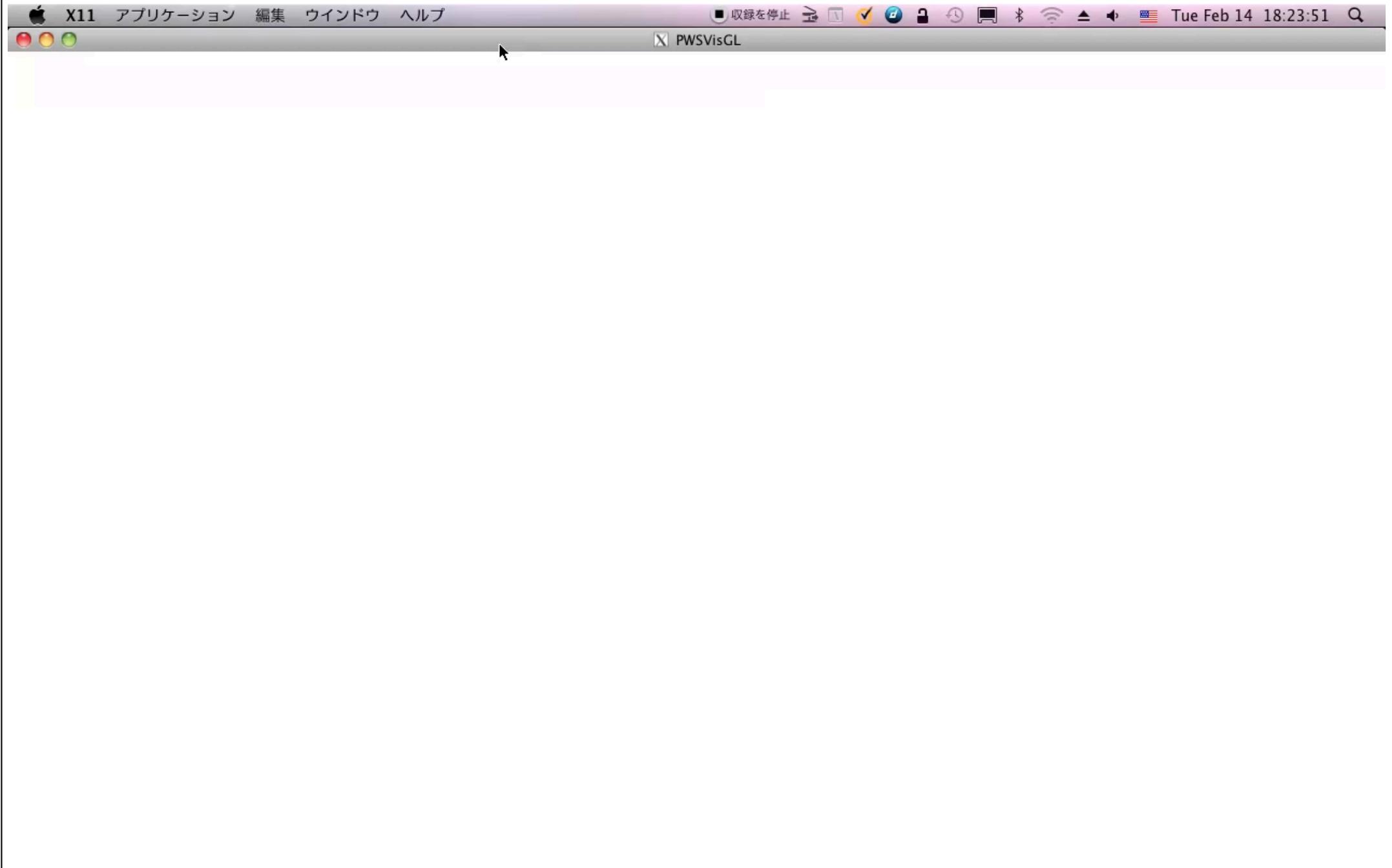
X線天文衛星のFITSデータの再生

- 逐次読み込みに対応したFITSライブラリの作成
 - ヘッダ情報の保持
 - 再生に必要なパラメータが集まっている
 - データは、現在読み取っている近辺(断片)のみ保持
 - 映像化に必要なデータのみ取り出し、他の要素は順次破棄
 - メモリ利用の効率化
- 汎用FITSライブラリを用いた場合と再生状況を比較
 - 今回は通信環境をPCで模擬
 - USB 1.1 カードリーダーからデータを読み込む
 - インタフェース速度: 12Mbps
 - 実効2Mbps
 - データ量: 8MB
 - 可視化要素: 観測時の天文衛星の位置

逐次読み込みの場合(実行画面キャプチャ)



汎用FITSライブラリを用いた場合（実行画面キャプチャ）



まとめ

- **モバイル環境を意識**
- **時系列データの可視化**
- **ユーザを待たせないインタフェース作り**
 - 逐次読み込み用FITSライブラリ
- **課題**
 - 技術面
 - 端末の処理能力
 - OS、機器のスペックの相違
 - 通信量・メモリの低減
 - 魅力的なコンテンツ