宇宙科学データの可視化
－モバイル環境に適した科学データの取り扱いについて－

三浦　昭[*1]　海老沢　研[*1]

Visualization of Space Science Data
~Science Data Handling Suitable for Mobile Devices~

Akira MIURA[*1], Ken EBISAWA[*1]

Abstract

We have been working on methods of visualizing space science data. For the purpose of education and public outreach, devices built on mobile computing platforms (iOS and Android, etc.) are fascinating ones to represent space science data. While mobile devices require a large amount of computing resources to visualize science data, wireless data services for mobile devices are not yet fast enough to instantly transfer the sufficient amount of data. This paper introduces a method to progressively download science data in order to reduce the latency. The proposed method here is also expected to reduce memory usage and CPU usage in comparison with conventional space science data I/O libraries.

Keywords: visualization, space science data, mobile platform

概　要

かねてより筆者らは宇宙科学データの可視化について検討してきた．最近の iOS や Android 等に代表されるモバイルデバイスは，教育・広報の観点からも無視できない存在になっている．科学データの可視化の 1 つの手段として，モバイルデバイスの性能は年々向上しているが，モバイル通信環境は未だ発展途上にあり，科学データを即時に転送できるまでには至っていない．本稿では，利用者を待たせることなく科学データを可視化するために，科学データを逐次取込みながら処理するための手法について提案する．本手法は従来手法と比較して，データ転送に伴う待ち時間を短縮するのみならず，デバイス内のメモリや CPU の使用率低減にも寄与することが期待される．

*1 宇宙航空研究開発機構宇宙科学研究所
(Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency)

1. はじめに

筆者らは，従来から JAXA が保有する宇宙科学データを広報や教育等の目的に供すべく，可視化や可聴化についての技術検討を行ってきた [1] [2] [3]．かような目的に適ったデバイスは，かつては PC であったが，昨今ではスマートフォンやタブレット等，モバイル型のデバイスが急速に普及しており，その勢力は無視できない規模となっている．本稿では，まず研究室や家庭の PC と比較して低速の CPU やモバイルデータ通信サービス，低容量のメモリ等，制約の多いモバイルデバイスにおいて科学データを取り扱う際の考察を行う．続いて，かような環境において，利用者の待ち時間低減やデバイスのメモリ効率向上を目的とした，科学データの逐次取込みと可視化の併用について提案する．本稿で提案する手法により，モバイルデバイスを利用した科学データの教育・広報が一層拡大することを期待するものである．

1.1 対象とするデータ

本稿では，「科学データ」の一例として，X 線天文衛星「あすか」や「すざく」の観測データを取り上げる．X 線天文衛星の観測データは，衛星に搭載された X 線望遠鏡が観測した個々の光子の飛来時刻や飛来方向等を時系列に保存して FITS[(4)] 形式にしたものである．以後本稿では，特に言及しない限り，「科学データ」は X 線天文衛星の観測データ（FITS 形式）を指すものとする．応用として，類似のフォーマットを有するデータに対しては，本稿で提案する手法が同様に適用できるものと考えられる．例えば筆者らが可視化・可聴化を試みた [3]PWS データもこの類である．

1.2 モバイルデバイスに纏わる動向

1.2.1 ハードウェア性能

iPod touch 登場以降のモバイルデバイスの高度化・高機能化は著しく，iPod touch から iPhone，iPad に連なる iOS デバイスや，各社が参入する Android デバイス等がしのぎを削っている．これらのモバイルデバイスは急速に普及しており，2011 年末時点の iOS デバイス累計販売台数は 3 億台以上 [1]，Android デバイスが 2 億 5 千万台以上 [2] と言われている．昨今の販売台数は，もはや PC を超えており [3]，携行用のデバイスとして主流になっていると言っても過言ではない．

最近の典型的なデバイスのスペック比較を，表 1 に示す．表中，CPU，RAM，解像度は主要家電メーカー及び携帯電話各社の公表値による．モバイルデバイスに搭載される CPU は，PC（デスクトップ機等）用とアーキテクチャが異なるため，同一 CPU クロックの最新 PC より性能は数段劣る．通信環境はモバイルデータ通信サービス（iOS，Android）及び有線のデータ通信サービス（家庭用 PC）について，関連する情報サイトの実測値 [4,5,6,7] を参考にした．なお無線アクセスポイントを用いる等，通信環境に配慮することによって，モバイルデバイスであっても，これ以上の通信速度を享受することは可能である．

表 1 各種デバイスのスペック比較（2012 年夏の例）

系統	形態	CPU [Hz] x コア数	RAM [Bytes]	画面解像度	通信環境
iOS	スマートフォン	800M x 2	512M	960 x 640	数 Mbps 〜
	タブレット	1G x 2	1G	2048 x 1536	数 Mbps 〜
Android	スマートフォン	〜 1.5G x 2 〜	1G 〜	〜 720 x 1280	数 Mbps 〜
	タブレット	〜 1.5G x 2 〜	1G 〜	〜 1280 x 800	数 Mbps 〜
家庭用 PC	据置	2G 〜 x 2 〜	2G 〜	1600 x 900 〜	数十 Mbps 〜
	ラップトップ	1.5G 〜 x 2 〜	2G 〜	1366 x 768 〜	数十 Mbps 〜

1 http://www.excite.co.jp/News/pc/20120217/Cobs_ip_201202_2011iosmac28.html
2 https://plus.google.com/+LarryPage/posts/jcyvVa5K4JW#+LarryPage/posts/jcyvVa5K4JW
3 http://japan.cnet.com/mobile/35013776/
4 http://www.musen-lan.com/speed/htmldata/
5 http://www.bspeedtest.jp/stat1_1.html
6 http://wimaxspeedmap.com/
7 http://mmd.up-date.ne.jp/news/detail.php?news_id=1014

PC の上位機種はこれらに比べると圧倒的な性能を実現しており，研究室等での解析・閲覧用途としてはモバイルデバイスと比べて PC が有利であることは言うまでもないが，家庭用等の低価格帯の製品を比較した場合，携帯会社の販売促進等の効果もあり，モバイルデバイスのコストパフォーマンスは決して悪くない．実際，ネットブックと呼ばれる低価格ノート PC に匹敵するスペックのデバイスも少なくない．とは言え，モバイルデバイスに搭載される CPU の性能やメモリ容量，バッテリ容量等は PC と比較して決して潤沢とは言えないので，大きな科学データを扱うアプリケーション開発においては，例えば高負荷の解析処理等，電力使用量が高くなる処理を避けつつ，同時にメモリ使用量も考慮する必要がある．

1.2.2　アプリケーション実行環境

モバイルデバイスと PC とでアプリケーションの実行環境を比較する．

PC においては，ブラウザの多機能化・高性能化に伴い，Flash や Java，JavaScript 等を用いた Web ベースのサービスが隆盛を極めている．これらのサービスは，ネイティブコードのアプリケーションに比べると，中間言語やスクリプティング言語を用いる分，性能面での不利益はあるが，PC の性能向上がそれを補っている．一方で PC 用のアプリケーションをインストールすることは，利用者がインストーラを能動的に取得した上で起動し，さらに幾つかの手間をかけることであり，一般の利用者にとっては敷居の高いものである．

これに対してモバイルデバイスでは，iPod の楽曲ダウンロード (iTunes) に始まる諸々のダウンロードサービスがアプリケーション配布にも拡大されており，様々なアプリケーションの取得やインストールは，とても簡易なものになっている．iOS 向けアプリケーションはネイティブコードであるため，ブラウザベースのサービスと比べて，ハードウェア性能を十分に発揮できる．Android ではハードウェア環境の互換性に重きを置いた Java アプリケーションと，性能に重きを置いたネイティブコードのアプリケーションと 2 種類の開発環境が提供されている．

1.2.3　通信環境

端末のハードウェア環境は着実に進化しているモバイルデバイスであるが，通信環境は必ずしも劇的な進化は遂げていない．家庭（建家）内の通信環境が ADSL から FTTH に進化し，インタフェース速度が 100M 〜 1Gbps になりつつあるのに対して，モバイルデータ通信サービスも高速の新商品が次々と発表されており，カタログスペックは数十 Mbps に至っているが，実測値は測定箇所によって様々である．1.2.1 節で参照した情報サイトに記された各地点の通信速度から推察するに，場所を限れば 10Mbps を超えるサービスを受けられる状況になりつつあるが，通勤や行楽等，広範囲のモバイル環境でサービスを受けられる速度としては数 Mbps 程度からであると考えられる．

すなわち，現在のモバイルデバイスは，処理能力は高くなっているが，それに供するデータを高速に取得することは必ずしも容易ではない状況にある．

そこで本稿では，低速のネットワーク環境においても利用者を待たせることなく科学データを取得し，可視化する手法の検討について述べる．

2.　モバイル環境における科学データの可視化

2.1　配信方法の比較（配信と可視化の順序）

モバイル環境で科学データの可視化を実現するにあたっては，大きく 2 つの方法が考えられる．ひとつは科学データを予め映像化して配信するものである．もうひとつは科学データそのものを配信してデバイス上で可視化するものである．執筆時点で iOS 向けのアプリケーションを“FITS”というキーワードで検索した結果，FITS 形式に対応していることを謳っているものは，フランスの線形加速器研究所（Laboratoire de l'Accélérateur Linéaire）が提供する ioda[8] と呼ばれるソフト 1 件のみであった．この ioda は，研究者向けも含めた種々のフォーマットのデータを表示できるソフトであるが，「あすか」や「すざく」のデータ形式には対応していないものであった．また表示できるデータはローカルのデータに限られるため，インターネット上のデータを表示するためには，事前にデータをダウンロード（もしくは PC からデバイ

8　http://ioda.lal.in2p3.fr/

スにデータをコピー）する必要がある．これに対して，YouTube 等の映像配信サービスはモバイルデバイスにも対応しており，科学映像も潤沢に提供されている．

2.1.1 映像を配信するメリット・デメリット

インターネット上の映像配信サービスは，視聴の待ち時間を短縮できるメリットがある．昨今の映像配信の手法は，大きく 2 種類に分けられる．一つはストリーミングであり，もう一つはプログレッシブダウンロードである．ストリーミングは映像配信を制御するセッション（RTSP として標準化されている）と映像のデータグラム（RTP として標準化されているが，これに限定されるものではない）で構成される．映像の再生・停止等の制御や，再生に必要なパラメータ等は RTSP で受け渡される．映像のデータグラムは，予めファイルに保存された映像でもリアルタイムに生成される映像でも配信可能である．映像が途切れることなく再生できるよう，端末側には幾ばくかのバッファが設けられている．端末側のアプリケーションによっては長時間の映像を保持できる場合もあるが，一般には映像全体のサイズに比べてバッファのサイズは僅かである．プログレッシブダウンロードは，予めファイルに保存された映像を配信する手法であるが，映像ファイルの冒頭に再生に必要なパラメータ類を集約することにより，ファイルの受信途中に映像を再生できるものである．受信したファイルは，再利用のため端末側に一時保存されることが多い．

比較的低速の通信環境でも再生できるよう，低解像度の映像から高解像度の映像まで選択できるようになっていれば，通信環境によらず映像視聴の待ち時間は低減できる．また異なるパラメータ設定で映像化したファイルを複数用意しておけば，様々な切り口での映像を視聴することも可能になる．

ただしこの方法は低ビットレートにしても，長時間の視聴に際しては通信量が大きくなる恐れがある．パラメータを変更して視聴する場合は，その都度異なるファイルを受信することとなるため，トータルの通信量は，さらに大きくなる．また映像を提供する側としても，視聴が想定される全科学データについて，予めシナリオを作成して映像化する必要があるため，映像化作業の負担は無視できない．

2.1.2 科学データを配信するするメリット・デメリット

科学データそのものを配信することにより，受信した端末上でデータを操作できるようになる．映像配信に比べて，一度受信したデータを用いて様々な切り口の可視化ができるため，複数の側面から可視化を試みる場合，トータルの通信量が低減できるメリットがある．最近のモバイルデータ通信サービスはパケット定額制を謳うサービスが主流となっているが，料金を定額にする範囲を限定するケースや，定額の代りに高速のサービスを受けられる通信量に上限を設けるケースが多く，トータルの通信量を低減することは利用者にとってもメリットがある．

ここで問題となるのが，科学データの受信に要する時間である．例えば 10MB 程度の科学データの可視化を考えた場合，研究室の LAN 環境や家庭の高速インターネットサービスであれば数十 Mbps 程度の通信環境が期待でき，受信にかかる待ち時間は数秒程度となる．これに対して，モバイルデバイスで想定されるような数 Mbps の通信環境では，受信に数十秒かかることとなる．一般ユーザとして数秒程度ならば待てるかも知れないが，数十秒の待ち時間は許容範囲を超えていると思われる．

2.2 科学データの速やかな可視化にとって望ましい手法とは

以上の点に鑑み，科学データをモバイルデバイスで可視化するにあたって望ましい手法を検討する．端末上でデータを操作でき，総通信量を低減できることを考慮すると，科学データそのものを配信して利用に供することが妥当と考えられる．しかしながら科学データを配信する方法では，長い待ち時間がデメリットとなることは前述の通りである．このような状況下で利用者を退屈させない方策，すなわち科学データ取得の間に「待っている」と認識させないような策を講じなければならない．

良く見られる対策としては，簡単なものは時計マークやプログレスバーで進捗状況を表示するものが多い．これは明らかに待ち時間であると宣言しているものではあるが，利用者が残り時間を把握できるという効果はある．その他，関連する情報を表示して，利用者が読んでいる間に処理が完了するもの等も見受けられる．

これに対して本稿では，映像配信の手法に倣い，受信途中にも可視化処理が可能な，「プログレッシブダウンロード」を科学データに適用することを検討する．検討項目は 2 点ある．ひとつは科学データを格納したファイルが適用条件を満たしているか，もうひとつは科学データを読み込むための既存ライブラリが条件を満たしているか，である．

2.2.1 「プログレッシブダウンロード」が適用できる条件

ここでは，一般に映像配信で用いられるプログレッシブダウンロードが本稿で取り上げる科学データにも応用可能であるか検討する．ここで実現を目指すのは，科学データを受信しながらの可視化処理である．映像配信サービスにおけるプログレッシブダウンロードを参考に，科学データを格納したファイルの適合条件を挙げると以下のようになる．

(a) ファイルの中で，可視化に必要なパラメータが冒頭にある．
(b) ファイルの中で，可視化に必要なデータが時系列に並んでいる．
すなわち，データ取込みの順に可視化できる．

X 線天文データを格納する FITS ファイルのデータ構造（「あすか」や「すざく」の例）を表 2 に示す．可視化に必要なパラメータ類は, ファイルの冒頭にあるプライマリ HDU とエクステンションのヘッダに記されている. それに続くデータ部には，観測された X 線の個々の光子についての情報（飛来時刻，飛来方向，エネルギーに関する情報等）が，観測された時刻順に格納されている．

表 2　X 線天文データ（FITS ファイル）のデータ構造

Header & Data Unit の種類	構成要素	備考
プライマリ HDU	ヘッダ	ファイル全般の情報
エクステンション （観測データがここに置かれる）	ヘッダ	観測データに関するパラメータ類
	データ	時系列の観測データ
エクステンション	ヘッダ	補助的なデータ類
	データ	

このフォーマットで作成されたファイルのヘッダ情報（可視化に必要なパラメータ類）を読み取った後，データが格納された順に個々の光子に関する情報を可視化すれば，天文衛星が観測した時の状況を時刻に沿って再現できる．すなわち，このフォーマットは前述の，プログレッシブダウンロードが適用できる条件を満たしている．また全データが揃わなくても，個々の光子の飛来方向（X-Y 方向）等に基づいて，ある期間の観測データを可視化でき，また，その輝度変化を再現する事もできる．極限すれば，この形式の FITS ファイルは，特殊なフォーマットの動画ファイルであるとも言える．

2.2.2 FITS ファイルの取込み

科学データを「プログレッシブダウンロード」するためには, それに適合した FITS ファイルの読込み処理が必要である．従来から提供されている FITS ライブラリ（FITS ファイルの読み書きのためのライブラリ）[(4)(5)] はデータの完全性を保証する構造となっており，ファイル全体を読込んで初めてデータの取扱いが可能となる．従来型 FITS ライブラリのデータ取込み動作を図 1 に示す. 図中ではデータ長を N としている. 左端が観測データを表しており, 従来型 FITS ライブラリは, 先ずこのデータを一括解釈する．その後に個々のデータにアクセスして，可視化することが可能となる．

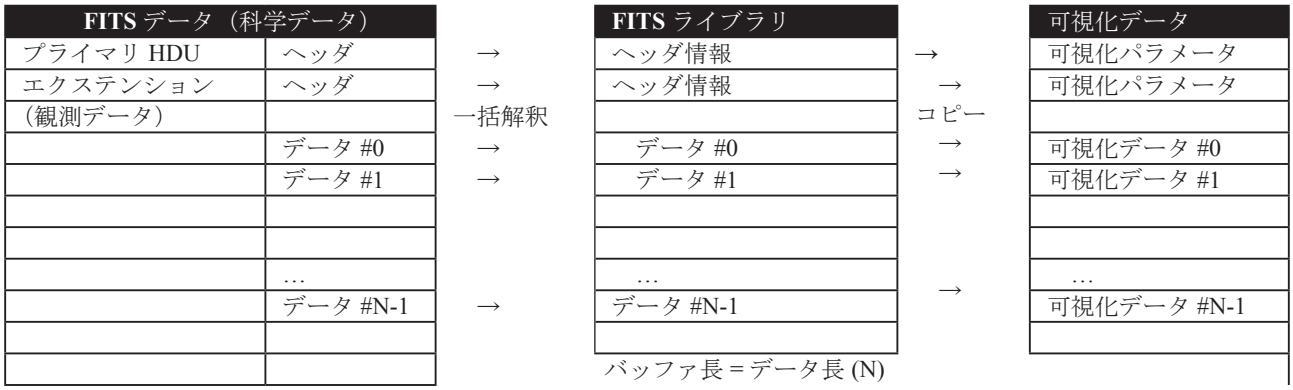


図 1　従来型 FITS ライブラリのデータ取込み動作

これは研究者等に対してデータの完全性を保証する上では妥当な設計であるが，FITS ファイルを読込み次第可視化するには適さない．また FITS データ取込み用にライブラリが作成するバッファのデータ長は，オリジナルの FITS ファイルのデータ長と同等となり，さらには，読み取ったバッファの中には，可視化に必要のない要素も含まれている．一時的とは言え，可視化処理中には，このような大きなバッファと，それに基づいて作成される可視化データと，二重にメモリを消費することとなる．限られたリソース上で動作するモバイルデバイスへの適用を考えた場合，従来型 FITS ライブラリには，メモリの利用効率と，FITS データ読込み完了までの待ち時間という 2 つのデメリットがある．このように従来型 FITS ライブラリでは「プログレッシブダウンロード」は実現できず，新たにライブラリを開発する必要がある．またメモリの利用効率の観点では，小規模のバッファを用いて映像を逐次再生するストリーミングの手法にも見習うべきものがあると考えられる．

3. FITS 逐次取込み型データ処理

以上の点に鑑み本稿では，メモリの効率的な利用と，科学データ読込み完了までの待ち時間短縮の観点から，FITS 逐次取込み型のデータ処理を提案する．提案手法の要点は，次の 2 つである．一つは，科学データを到着順に取込み，逐次可視化処理に供することができることである．もう一つは，ライブラリとしてのバッファ領域を必要最小限にし，メモリの利用効率を高めることである．これにより，モバイルデバイスでの低速ネットワーク環境における待ち時間短縮と，限られたメモリの有効利用が期待できる．

3.1 処理の流れ

データ処理の過程を図 2 に示す．図中，リングバッファ長 n はデータ長 N より十分小さな値としている．この処理過程においては，まず FITS データの冒頭からヘッダ情報を読み取り，可視化に必要なパラメータと，後続のエクステンション (binary table extension) に格納された観測データの構造（レコード毎の値の並び及びそれらの型・サイズ）を抽出する．次に，観測データの構造に基づいて，各データを読み取るためのリングバッファ，及び，リングバッファから観測データ構造に即して値を解釈するためのメソッド群を構築する．続いて，エクステンション中の観測データを逐次読み取る．リングバッファの入力はエクステンションから読み取った各レコードのバイナリ値である．これらの値は，バイトオーダーの変換以外の解釈はせずにリングバッファに格納される．リングバッファに短期的に取込まれている間に，可視化に必要な要素のみを可視化に適した形式に変換して保持する．すなわち，可視化に不要な観測データは，リングバッファにバイナリデータとして転記されるのみであり，値を解釈するオーバーヘッドが低減される．リングバッファは元の観測データ長に比べて十分短く，バッファ中の情報は取込みの過程で新しい観測データに上書きされるため，従来型の FITS ライブラリと比べて，読み取り時に占有するメモリ量を控えることができる．

可視化データへの変換処理は，取込まれた情報がリングバッファ中に維持されている期間内に実行するため，可視化データは観測データ取込み開始早々から順次利用可能となり，取込み完了までの待ち時間の間にも，可視化データをディスプレーに表示することが可能となる．

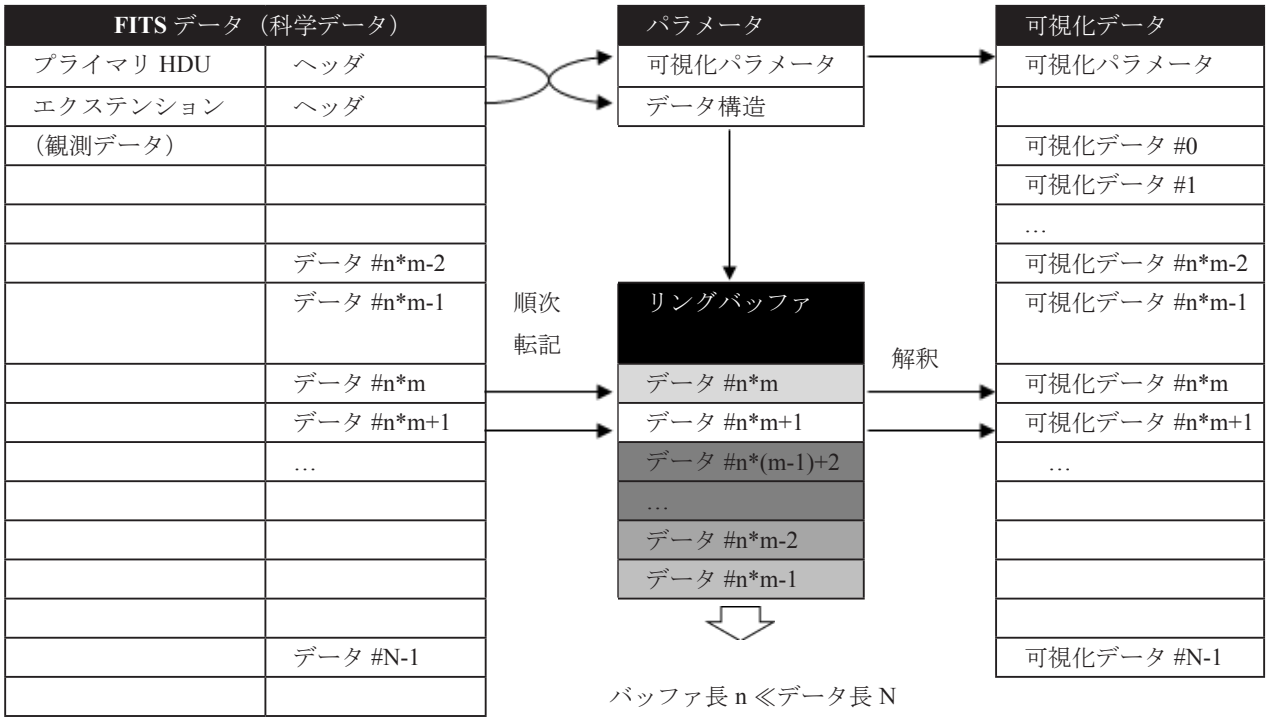


図 2 FITS 逐次取込み概要

3.2　対象となるデータ型

FITS データを逐次取込む都合上，FITS フォーマットで定義されているデータ型の内，一部は提案手法の適用対象外とする．本手法において，binary table extension 中の対応可能なデータ型を表 3 に示す．

表 3　FITS 逐次取込み型データ処理の適用対象

データ型	適用対象	内部表現
Logical	○	char
Bit	○	char
Unsigned byte	○	uint8_t，int8_t
16-bit integer	○	int16_t, uint16_t
32-bit integer	○	int32_t, uint32_t
64-bit integer	○	int64_t, uint64_t
Character	○	char
Single precision floating point	○	float
Double precision floating point	○	double
Array Descriptor (32-bit)	×	適用外
Array Descriptor (64-bit)	×	適用外

Binary table extension に定義されている型の内，Array Descriptor は後続のデータ領域中に格納された値を指し示すポインタであり，短いリングバッファでは取込んだ順に値を解釈することが困難である．長いバッファを使用することは，本稿で課題としている待ち時間の短縮やメモリ使用効率の向上とは相反するものであり，このような型を使用する FITS データは，提案手法の効果が薄いと判断される．幸いにして今回検討対象とした X 線天文衛星「あすか」や「すざく」の FITS データでは使用されていない型でもあり，Array Descriptor (32-bit)，Array Descriptor (64-bit) 共に，提案手法の適用対象外とした．

4. 模擬環境による比較

4.1 模擬環境の条件

本稿では，提案手法と従来手法との比較を以下の模擬環境下で行った．

(a) 開発環境 :Mac Pro (Early 2008)，Mac OS X，Xcode
iOS ベースのモバイルデバイスとの開発環境共有が可能であるため，Mac OS X と Xcode を基本としたプログラミングを行っている．Android は複数の CPU アーキテクチャが混在するため，互換性を確保するためには Java に基づいたアプリケーション開発が必要となるが，CPU 毎のネイティブコードで開発できる環境も整いつつあり，Xcode で開発したソースの共用も可能になると考えられる．

(b) 通信環境模擬 : USB 1.1 カードリーダ＋ SD カード
低速のデータ通信を，USB 1.1 経由のファイル読み込みで模擬した．USB 1.1 の通信速度は 12Mbps であるが，模擬環境下でのファイル読み込みの実測値は約 3Mbps 程度であった．この速度は，筆者の執務室におけるモバイルデータ通信サービス（UQ WiMAX）の利用環境と比較すると，実際の通信速度に近いものであるが，さまざまなモバイルデータ通信サービス全般を近似できるものではない．

(c) 従来手法で用いたライブラリ : sfitsio[4]
C++ でコーディングされた FITS ライブラリである．C++ の機能を用いて分かりやすいインタフェースが提供されている．C++ の高度な知識は要求せず，伝統的な CFITSIO と比べて利便性の高いライブラリとなっている．

(d) 可視化ライブラリ : Open GL と C++ を用いた独自開発のライブラリ
Android や iOS との共用ができるよう，Open GL ES (Android や iOS で採用されているバージョン) との互換性を考慮している．

(e) 取込み対象の FITS データ :「あすか」が観測した，かにパルサーのデータ
DARTS で提供されている FITS データの中から，「あすか」衛星搭載の GIS 観測装置が 1995 年 9 月 15 日に取得した ad10405000g300170h.evt (10.6MB) を用いた [10].

(f) FITS データ取込みと可視化手順 : プログラム起動直後に軌道要素と FITS データの取込みを開始する．取込んだ FITS データは，従来手法・提案手法それぞれにおいて，ライブラリの FITS データ取得関数が戻り次第可視化を開始し，全データ取得が完了するまで，観測データの可視化データへの変換と，変換結果の描画を繰り返すものとする．従来手法においては，FITS データ取得開始から完了までが 1 回の処理となるため，実際に繰り返しの処理が発生するのは，提案手法のみとなる．

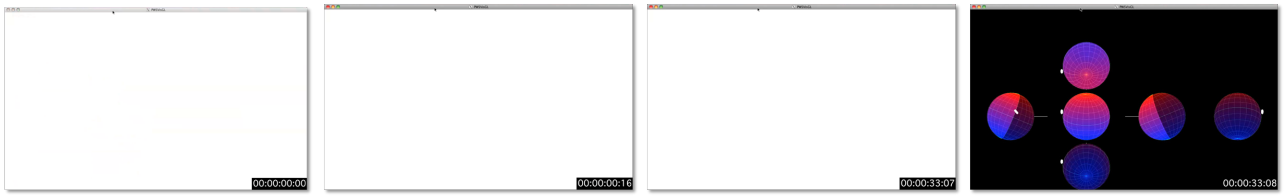
(g) 可視化する項目 : 対象天体（かにパルサー）観測時の「あすか」の軌道
観測期間中に刻々と変わる「あすか」の軌道上の位置をプロットする．位置計算に必要となる軌道要素は，上記 FITS データと共に DARTS で提供されている
http://darts.jaxa.jp/pub/asca/data/10405000/aux/frf.orbit.204.gz
を用いた．なお FITS データ取込みと並行して可視化できる項目は，これに限定されるものではなく，一般に時系列に可視化できるものは，補助的なデータ（キャリブレーション等）が必要となる場合があるものの，原則として FITS データ取込みと並行して可視化可能である．
閏秒については，現時点ではプログラムのソースに補正項目を埋め込んでいる．

4.2 模擬環境における実行結果

実行画面を図 3 に示す．上段が従来手法の実行結果，下段が提案手法の実行結果である．それぞれ左から右へ時系列に示す．各シーンの下の数字は，プログラム起動時からの経過時間（分 : 秒 ; フレーム）である．画面が空白となっているシーンは，可視化に要する最初の観測データが取込みルーチンから戻ってくるのを待っている時間帯である．6 つの球体が描かれているシーンは，取込んだ観測データに基づいて，その時刻の衛星の位置を描画したものである．球体が地球を表し，6 方向から見た衛星の位置を白丸でプロットしている．

10 ftp://ftp.darts.isas.jaxa.jp/pub/asca/data/10405000/screened/ad10405000g300170h.evt.gz

従来手法



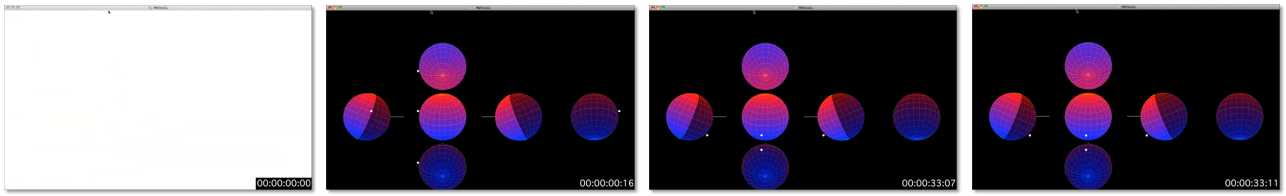
00:00;00
起動

00:00;16
観測データ取込み開始

00:33;07
観測データ取込み中

00:33;08
観測データ取込み終了
位置情報の描画が可能に

提案手法



00:00;00
起動

00:00;16
観測データ取込み開始
位置情報の描画が可能に

00:33;07
観測データ取込み中
継続して位置情報更新中

00:33;11
観測データ取込み終了
位置情報の更新完了

図 3 従来手法と提案手法の実行結果

プログラム起動からデータ取込み終了までの時間は，従来手法・提案手法共に 33 秒程度となり，殆ど差がなかった．従来手法ではこの 33 秒余りの間は取得途中の観測データを参照できず，衛星の位置情報も描画できない．提案手法においては，0 秒の 16 フレーム目（約 0.5 秒目）から衛星位置の描画が可能となっており，この時点から取込みが完了する 33 秒の 11 フレーム目までの間は，取得途中の観測データから抽出した観測時刻情報に基づいた衛星の位置をアニメーションで描画できている．これが実際の科学データ可視化アプリケーションであれば，利用者にとっては，30 秒余りの観測データ取得時間は待ち時間と認識されなくなると期待される．

5. おわりに

モバイルデバイスを前提としたデータ通信サービスにおいても利用者を待たせることなく科学データを可視化する手段として，FITS データの逐次取込みについて提案した．この手法の利点は，低速の通信環境においても利用者の待ち時間を低減できることや，デバイス側のメモリ消費量を低減できること等が挙げられる．また高速ネットワークを利用している場合であっても，ネットワークの輻輳などに起因する通信速度の低下に対しては同様の効果が期待できる．今回比較に用いた科学データは 10MB 余りと，昨今のモバイルデバイスの記憶容量と比較すると十分小さいとみなすこともできるが，観測によっては 100MB を超えるデータもあり，通信環境やデバイスの処理能力等が限られる中では，従来手法と比べて提案手法を用いた可視化は適用できる科学データの範囲が広がるものと期待される．

一方で従来の FITS ライブラリが，一旦取込んだデータは，明示的に消さない限り，継続してアクセスできるのに対して，提案する手法の場合，デバイス上にバッファされているデータは限られており，開発者はバッファされているデータの範囲を常に意識しながらプログラミングしなくてはならない．また提案手法は科学データのフォーマットに即した取込み処理を新規に作り上げることとなり，旧来から実績のあるライブラリに比べて，データの再現性等の信頼性を確立するには，尚一層の検証が必要であると考えられる．ただし信頼性の検証については，従来型の FITS ライブラリを用いた場合も，ライブラリ開発者が想定していなかったモバイルデバイスに移植する際には，適切なコンパイルがなされているか逐一検証する必要があり，動作検証の労力は提案手法のみに発生するものではない．

今後は提案手法の動作検証を重ねると共に，同手法を用いたモバイルデバイス上での可視化について応用例を積み重ねて行く必要がある．また教育・アウトリーチの他に，研究者向けの応用も考えられる．詳細な解析に用いるためには

デバイスの性能や容量等とのトレードオフが必要であるが，モバイルデバイスを用いた衛星運用状況のモニターや観測データの Quick Look 等，軽快なレスポンスを必要とする用途には適用可能であると思われる．

参考文献

1. 大規模宇宙科学データの特徴量抽出と映像・音声化 - 教育利用として -. 三浦昭，海老澤研 . 2011 年 , 日本天文学会 2011 年春季年会予稿集 .
2. 宇宙科学データの時系列可視化と携帯型デバイスへの応用検討 . 三浦昭，海老澤研 . 2012 年 , 日本天文学会 2012 年春季年会 .
3. 宇宙科学データの可視化・可聴化 : 教育・広報利用 . 三浦昭 , ほか . 2012 年 , 宇宙科学情報解析論文誌 , ページ : 13-22.
4. *SFITSIO - A Next-Generation FITS I/O Library for C/C++ Users.* **Yamauchi, C.** 2010, Astronomical Data Analysis Software and Systems XIX, Vol. 434, pp. 469-472.
5. *Definition of the Flexible Image Transport System (FITS), version 3.0.* **W. D. Pence, L. Chiappetti, C. G. Page, R. A. Shaw and E. Stobie.** A42, 2010, Astronomy and Astrophysics, Vol. 524.
6. *CFITSIO, v2.0: A New Full-Featured Data Interface.* **Pence, William.** 1999, Astronomical Data Analysis Software and Systems VIII, Vol. 172.