



ISSN 1349-1113
JAXA-RR-15-006

宇宙航空研究開発機構研究開発報告

JAXA Research and Development Report

宇宙科学情報解析論文誌 第五号

2016年3月

宇宙航空研究開発機構

Japan Aerospace Exploration Agency

序文

宇宙科学情報解析論文誌(Journal of Space Science Informatics Japan)第五号をお送りします。

近年、宇宙科学における様々な分野において、観測装置と計算機の大型化によって、以前に比べて飛躍的に大量のデータを取得することが可能になってきています。それによって、観測から優れた科学的成果を産出するためには、大規模データをいかに効率良く解析し、必要な情報を引き出すかが重要な技術課題となりつつあります。また、デジタル技術の普及に伴い、デジタルプラネタリウムや様々なアプリケーションに代表されるように、宇宙観測データが科学研究以外の分野でも広く利用されるようになってきています。

そのような状況を鑑み、JAXA 宇宙科学研究所・学際科学研究系では、その前身である宇宙科学情報解析研究系、宇宙科学情報解析センター(PLAIN センター)の時代より、主に JAXA の科学衛星・探査機によって得られた宇宙観測データを収集し、広く国内外の研究者による利用を促進するためのサイエンスデータベースの開発・運用を行うとともに、新たなデータ解析手法の研究等を行ってきました。今後は、学際科学研究系を中心に複数の研究系に属するメンバーが、宇宙科学データ(シミュレーションデータを含む)に関する新しい処理・解析・利用の手法、データベース技術やそれを応用したシステムの構築・運用技術など、宇宙科学・情報科学・情報技術に関わる分野横断的な成果発表および情報交換の場として「宇宙科学情報解析シンポジウム」を開催するとともに、それに関連した研究論文を集めた「宇宙科学情報解析論文誌」を発行して参ります。当論文誌が、周辺領域の研究者、技術者間の情報交換や活発な議論につながることを期待しています。

2015 年 12 月

JAXA 宇宙科学研究所

学際科学研究系

高木 亮治, 三浦 昭, 山本 幸生

太陽系科学研究系

篠原 育

宇宙物理学研究系

海老沢 研

plainsympo@c-soda.isas.jaxa.jp

目次

衛星運用

「超小型衛星の運用に適したネットワーク対応地上局ソフトウェアの設計と実装」
堀口 淳史, 橋本 論, 久保田 晃弘…………… 1

「第二世代 EDISON (衛星運用工学データベースシステム) の開発」
高木 亮治, 北條 勝己…………… 23

データ処理、データ解析

「レベル1時系列データフォーマット変換ツールの開発」
松崎 恵一, 山本 幸生, 高木 亮治, 篠原 育…………… 29

「超小型位置天文衛星 Nano-JASMINE のデータ解析」
山田 良透…………… 43

「月周回衛星『かぐや』のデータを用いた多目的最適化による月着陸最適候補地の選定」
西山 万里, 大嶽 久志, 星野 健, 橋本 樹明, 渡辺 毅,
立川 智章, 大山 聖…………… 51

アーカイブ開発

「JAXA キュレーションセンターにおけるリターンサンプルデータの管理システムの
開発とその運用状況」
上梶 真之, 矢田 達, 唐牛 譲, 中藤 亜衣子, 熊谷 和也, 橋口 未菜子,
松本 徹, 岡田 達明, 安部 正真 …………… 59

「全天 X 線監視装置『MAXI』のデータアーカイブの開発」
中川 友進, 海老沢 研, 中平 聡志, 上野 史郎, 富田 洋, 木村 公,
杉崎 睦, 芹野 素子, 三原 建弘, 根來 均…………… 71

アプリケーション・ソフトウェア開発

「JavaFX-based iUgonet Data Analysis Software (JudasFX)のプロトタイプ開発」
小山 幸伸, 佐藤 由佳, 中野 慎也, 八木 学, 田中 良昌, 阿部 修司, 能勢 正仁,
蔵川 圭, 池田 大輔, 梅村 宜生, 新堀 淳樹, 上野 悟…………… 81

「地球惑星科学分野のクイックルックサービス」

今井 弘二, 海老沢 研, 池田 大輔, 北尾 大介…………… 93

「適切かつ高度なデータ処理環境を提供するためのウェブインターフェース

“Web Processing Catalog” かぐや搭載スペクトルプロファイラデータを事例として」

林 洋平, 山本 幸生, 小川 佳子, 平田 成, 出村 裕英,
寺藺 淳也, 松永 恒雄…………… 111

「教育用計算機の利用を想定した3次元データの簡便な可視化手法の提案—gnuplotの利用」

野口 克行…………… 123

データ可視化

「『はやぶさ2』運用補助のための可視化アプリケーション -影の描写の検討-」

三浦 昭, 山本 幸生, 吉川 真…………… 133

データ利用拡大、アウトリーチ

「JAXA OPEN APIによる地球観測データの利用拡大」

秋山 恭平, 可知 美佐子, 西原 雄次, 館下 博昭, 安部 眞史,
高井 元, 石井 宏宗…………… 149

「観賞性に重点を置いた『すばる』望遠鏡 HSC 画像の画像処理」

上坂 浩光…………… 159

超小型衛星の運用に適した ネットワーク対応地上局ソフトウェアの設計と実装

堀口 淳史^{*1}, 橋本 論^{*2}, 久保田 晃弘^{*3}

Design and implementation of the Networked Ground Station Software for the Operation of Nano Satellites

Junshi Horiguchi^{*1}, Ron Hashimoto^{*2}, Akihiro Kubota^{*3}

Abstract

This paper describes the design and implementation of the software named “artsatd”. It is a networked ground station software for the operation of 1U cubesat “ARTSAT1:INVADER” and micro spacecraft “ARTSAT2:DESPATCH”. It is optimally designed to operate nano satellite using amateur radio band. The “artsatd” program is a system resident daemon program which supports indispensable functions for the remote operation of satellite, such as getting a list of satellites and orbital elements from the internet, controlling and checking the ground station hardwares at remote places, cooperative operation of ground stations by multi-users and streaming received sound signal from the transceiver, etc. Since artsatd is a server-client style daemon software, it doesn't have the graphical user interface by itself. However, it has a graphical user interface written in HTML. It is also compatible with the remote procedure call as JSONRPC. By using the functions of JSONRPC, we can easily operate ground station using scripts and the connection between artsatd and other web applications. In the last section, we show some results of operations of INVADER and DESPATCH using artsatd.

Keywords: Art Satellite, CubeSat, Network, Ground Station, Programming, Software, C++

概要

1U CubeSat の芸術衛星「INVADER」と 50cm 立方の芸術宇宙機「DESPATCH」の運用に使用された、ネットワーク対応の地上局ソフトウェア「artsatd」の設計と実装の詳細について述べる。artsatd は、アマチュア無線帯を利用した超小型衛星の運用に最適な設計を目指しており、衛星一覧や軌道要素をインターネットから取得する機能に加えて、地上局のハードウェアを遠隔地から操作・監視する機能、複数人数での共同操作機能、無線機で受信した音声信号のストリーミング機能などを備えるシステム常駐型のデーモンソフトウェアである。サーバ・クライアント型のデーモンソフトウェアであるためグラフィカルユーザインターフェースを持たず、HTML で記述されたユーザインターフェースや、JSONRPC を使用したリモート関数呼び出しの機能を備える。JSONRPC による地上局の操作機能により、スクリプトを使った自動操作や外部のウェブアプリケーションとの連携を柔軟に行うことができる。最後に、artsatd を使用した INVADER 衛星と DESPATCH 宇宙機の運用の成果について報告する。

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

*¹ 多摩美術大学 × 東京大学 ARTSAT PROJECT (Tama Art University x Tokyo University ARTSAT PROJECT)

*² 多摩美術大学 (Tama Art University)

*³ 多摩美術大学 (Tama Art University)

1 はじめに

超小型衛星を打ち上げ運用するためには、超小型衛星の開発と同時に地上局の準備を進める必要がある。超小型衛星の運用に必要な地上局のシステムは比較的単純なものであるが、アンテナ・無線機・モデムなどのハードウェアの設置と軌道計算を行ったり、ハードウェアを制御するためのソフトウェアの準備が必要となる。

アマチュア無線帯を利用した超小型衛星プロジェクトでは、プロジェクト単位で専用の管制ソフトウェアを開発したり、有志のソフトウェア作家が作成した軌道計算ソフトウェア^{*4}とアンテナや無線機の制御ソフトウェアを連携させた運用が行われている。衛星の追尾など運用の根幹に関わる部分において、複数のソフトウェアを連携させて動作させることは、さまざまな人為的ミスを起こす原因となる。反面、運用情報の公開など付随的な部分では、ソフトウェアの連携は非常に有用である。そこで、ハードウェアの制御を統合的に行い、インターネット対応など高度なレベルで操作できるソフトウェアを複数のプロジェクトで継続的に利用できると便利である。

ARTSAT プロジェクトでは INVADER 衛星^{*5}の管制を行うために artsatd と名付けられた地上局ソフトウェアを開発した【図1】【図2】。artsatd は設計の段階から複数のプロジェクトで利用されることを想定しており、続く DESPATCH 宇宙機^{*6}の運用では惑星軌道の計算にも対応し、設定されたミッションを達成した。継続的な利用を想定しているためオープンソースとして開発^{*7}し、アマチュアから専門家まで幅広い利用に対応できるように設計されている。

本論文では、超小型衛星の運用に適した地上局ソフトウェアである artsatd の設計と実装の詳細について述べる。

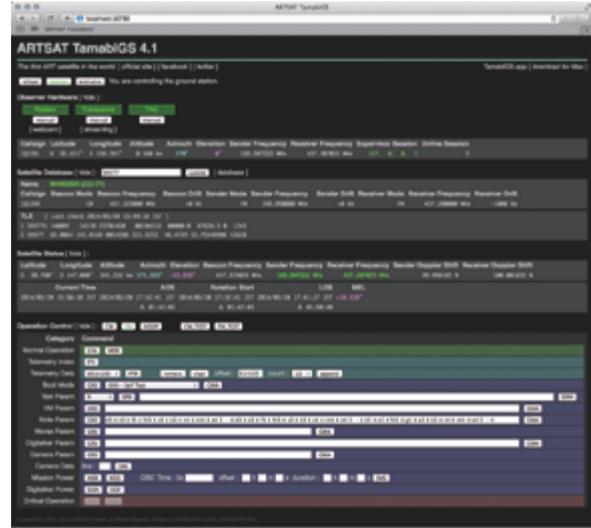


図1 地上局ソフトウェア artsatd



図2 多摩美地上局と artsatd (iMac 内)

2 一般的な地上局の構成と必要とされる機能

2.1 ハードウェアの構成

一般的に超小型衛星を運用するためにはアンテナ・無線機・モデム (TNC)・パーソナルコンピュータが最低限必要となる。アンテナは、アマチュア無線帯を送受信できるものに方位角と仰角を変えられるようにローテータを取り付ける。無線機は、アマチュア無線帯を送受信することができ、シリアル通信などを使用して外部から周波数を制御できることが望ましい。モデムとパーソナルコンピュータは、衛星と FM パケッ

^{*4} Orbitron - <http://www.stoff.pl>

^{*5} NORAD 39577, コールサイン JQ1ZKK, オスカーナンバー CO-77

^{*6} NORAD 40321, コールサイン JQ1ZNN, オスカーナンバー FO-81

^{*7} http://github.com/ARTSAT/ground_station

ト通信を行う場合に必要である¹⁾。また、これらの他にアンテナで受信した電波を増幅させるためのプリアンプや受信した音声信号を録音するための録音機、アンテナの状態を監視するためのウェブカメラなどがあると理想的である。

一例として、ARTSAT プロジェクトでは【表1】のハードウェアを多摩美術大学情報デザイン学科に開設した多摩美地上局に設置している。これらのハードウェアは【図3】のように接続されている。

表1 多摩美地上局のハードウェアの構成

種別	機種
送信アンテナ	M2 Antenna 2MCP22
受信アンテナ	M2 Antenna 436CP42UG
プリアンプ	AG-35
ローテータ	YAESU G-5500 + YAESU GS-232B
無線機	ICOM IC-9100M + ICOM CT-17
安定化電源	ICOM PS-126
モデム	TASCO TNC-555
ウェブカメラ	CEPSA CPB643W
録音機	ZOOM H4n
統合制御	Apple iMac + OS X 10.9.5

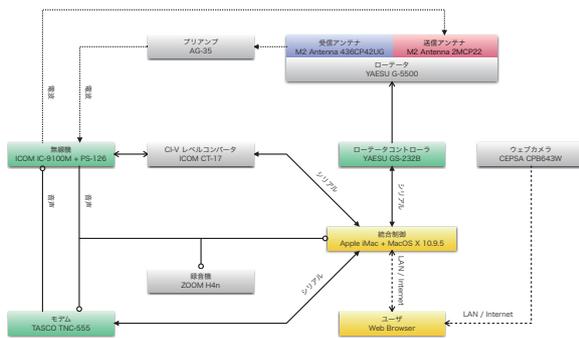


図3 多摩美地上局のハードウェアの接続

2.2 運用に必要な機能

2.2.1 軌道要素の取得

地上局から衛星を管制するためには、衛星の現在位置と移動方向を知る必要がある。これらを知るためには、NORAD と NASA が提供している衛星の 2 行軌道要素形式 (TLE) を利用する

方法がある。最新の TLE は CelesTrak^{*8} のウェブサイトから取得できる。

2.2.2 軌道の計算

次に SGP4 / SDP4 アルゴリズムや SGP8 / SDP8 アルゴリズムを用いて、取得した TLE から衛星の現在位置と移動方向を計算する。TLE には観測した時刻を基にした元期が設定されており、元期から時間が経過するほど計算精度が悪くなる。そこで、なるべく新しい TLE を用いた軌道の計算が必要となる。

2.2.3 方位角・仰角の追尾

衛星の現在位置と地上局の位置関係から衛星の方位角と仰角を計算できれば、それらの情報を用いてアンテナを衛星に向けることが可能となる。小型の八木アンテナであれば手で持って衛星の方向に向けることもできるが、実際の運用では現実的ではない。大型のアンテナの場合にはローテータをローテータコントローラを介して連続的に制御する必要がある。

2.2.4 周波数の追尾

衛星は高速で移動しているため、無線周波数にドップラーシフトが発生する。ドップラーシフトの割合は衛星の現在位置と移動方向、地上局の位置関係から計算することが可能であり、ドップラーシフトを考慮した周波数で衛星と通信を行わなければならない。ドップラーシフトの割合は常に変化するので、継続的に計算を行い周波数を補正する必要がある。

2.2.5 コマンドの送信

衛星を運用するには、衛星が可視範囲に存在するときにアンテナを衛星に向けて、ドップラーシフトを考慮した周波数に無線機を設定して通信を行う。アマチュア無線帯を利用する衛星では、FM パケット通信を使用してコマンドの送信を行うことが多い。ユーザインターフェースから入力されたコマンドは、デジタル信号を音声信号に載せるために TNC を使った変調を行い、無線機から出力する。

*8 <http://celestrak.com>

2.2.6 データの受信

衛星から常時送信されている CW ビーコンや FM パケット通信での応答を地上局で受信する必要がある。送信時と同じくアンテナを衛星に向けて、ドップラーシフトを考慮した周波数に無線機を設定する。地上局では受信された音声信号を録音しておき、運用後の検証作業に役立てる。FM パケット通信を受信した場合には、TNC を使って音声信号を復調しデジタル信号を取り出す。復調されたデータを地上局のデータベースに時系列に保存する。

3 アマチュア利用に適した機能

3.1 効率的な衛星運用

アマチュア無線帯を利用した超小型衛星を大学生の学習的プロジェクトとして運用する場合や、アマチュア無線家が趣味として運用に参加する場合、衛星の打ち上げから運用停止までのあいだ毎日地上局で運用を行うことは現実的ではない。衛星を運用する時刻は投入される軌道に依存⁹⁾し、毎日同じ時刻にやってくる衛星もあれば季節によってずれてゆく衛星も存在する。限られた人的リソースを用いて日々の運用を行なっていくには、地上局の自動化や遠隔操作への対応は必須の要件となる。

前章で述べたように、地上局における基本的な機能や動作はいくつかの項目に整理することができる。それぞれの項目に応じてどのように自動化できるのか、どの部分が遠隔操作できると効率的なのかを検討する。例えば、軌道要素の取得を手動で行う場合、ウェブサイトを開いて最新の TLE を探し、TLE が更新されていたら地上局ソフトウェアに入力するという運用ごとに行う必要がある。これは大変手間のかかる作業であり、またデータを見間違えるなどの人為的ミスが犯す可能性が高まる。そこでウェブサイトをクロールして TLE の更新時刻を自動的に判断し、更新されていたら既存のデータを置き換えるという作業を自動化できると効率的である。

アマチュアによる運用では、自動化されたソフトウェアがなんらかの理由でエラーを起こし

動作しないデメリットよりも、自動化することで得られる人的リソースのメリットの方が大きい。また、致命的な事象へと発展する可能性が低いことから積極的に自動化することを考える。アマチュアであっても一度限りの機会しかない運用や絶対に失敗が許されない操作の場合には、ソフトウェアによる自動化だけに頼らず、あえて人間が操作を行うことも重要である。

3.2 地上局のネットワーク対応

地上局の自動化や遠隔操作への対応を行うには、地上局のネットワーク対応が必須の条件となる。地上局をネットワーク対応することにより、最新の軌道要素をウェブサイトから自動的に取得したり、遠隔地から地上局を操作したり監視したりすることが可能となる。

地上局をネットワーク対応させると次のような項目が実現できる。

- 最新の軌道要素の自動取得
- 遠隔地からの地上局の操作や監視
- 複数人での地上局の共同操作
- ウェブカメラを用いたアンテナの監視
- 無線機で受信した音声信号のストリーミング
- 受信データのリモートデータベースへの送信
- 外部ソフトウェアからの地上局の状態の取得

3.3 地上局システムの移植性

アマチュア利用においては、完成した地上局のシステムを他のアマチュアも簡単に複製できると便利である。高度に専門的な機材やソフトウェアをなるべく使用せず、誰でも簡単に手に入る機材でシステムを構成することにより、同じシステムの複製版や移植版が数多く誕生しシステムの信頼性向上につながる。

地上局のハードウェアは、アマチュアが通常の購入ルートを通じて入手できる機材を利用し、ソフトウェアは POSIX / BSD 準拠として、特定のハードウェアや OS に依存しない実装を行うとよい。

⁹⁾ INVADER 衛星は、高度 378 km、傾斜角 65 度の太陽非同期軌道に投入され、季節に応じて運用時刻が遅くなった

4 ソフトウェアの設計と仕様

4.1 アプリケーション vs. デモン

地上局ソフトウェアをネットワーク対応させる場合にはいくつかの手法が考えられる。1つ目はネットワーク対応が考慮されていない GUI アプリケーションを遠隔地から操作する場合であり、RealVNC^{*10} などのリモートデスクトップソフトウェア^{*11} を使うことで遠隔地に居ながら使い慣れた地上局ソフトウェアを使用することができる【図4】。X ウィンドウシステムを使用したアプリケーションの場合も同様である。

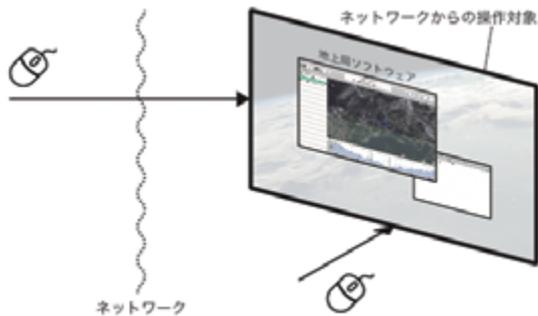


図4 VNC を使用した遠隔地からの操作

2つ目は GUI アプリケーションでありながらネットワークを介した専用コマンドを受け付け、主要な操作ができるアプリケーションを使用する方法である【図5】。

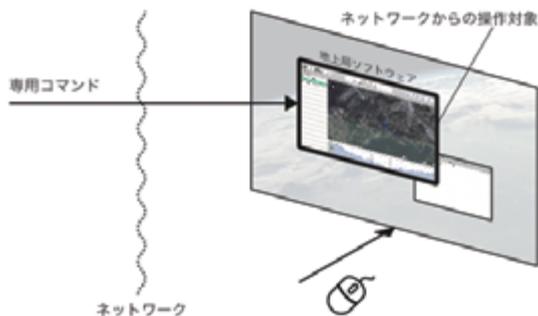


図5 専用コマンドを使用した遠隔地からの操作

3つ目は GUI を持たずネットワークを介したアクセスのみに応答し、システムに常駐して動作するデーモンソフトウェア^{*12} を使用する方法である【図6】。

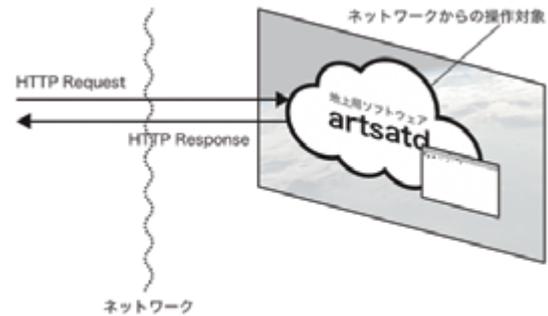


図6 デーモンを使用した遠隔地からの操作

1つ目の方法は使い慣れた環境をそのまま遠隔地から操作できるため、すでに地上局を運用している場合には有用な手法である。しかし、使用しているソフトウェアはネットワーク対応について設計の段階からは考慮されていないため、ネットワーク対応の恩恵を最大限受けることはできない。

1つ目と2つ目の方法は GUI アプリケーションを使用するため、地上局の運用中はアプリケーションが必ず起動されている必要がある。他の GUI アプリケーションと基本的には同じ設計であり、ユーザが間違えてアプリケーションを終了させるといような不注意による操作ミスの危険性も高い。

地上局ソフトウェアはその性質上、システムの起動から終了まで常に衛星の軌道を計算して、ハードウェアを制御することが求められる。また、遠隔地から任意の時点でアクセスして地上局を操作できるようにするという事は、複数人での同時アクセスに対応しなければいけないことを表している。

以上のことを考慮すると、地上局ソフトウェアをサーバ・クライアント型の設計とし、システムにデーモンソフトウェアとして常駐させることは理にかなっている。ARTSAT プロジェク

^{*10} <http://www.realvnc.com>

^{*11} X Window System, Virtual Network Computing, Apple Remote Desktop などが相当する

^{*12} UNIX のバックグラウンドプロセスである syslogd, sshd, ftpd などが相当する

トではこのような常駐型の地上局ソフトウェアを artsatd と名付ける。【表 2】にアプリケーションとデーモンの比較をまとめる。

表 2 アプリケーション vs. デーモン

	アプリケーション	デーモン
起動	ユーザ操作	システム起動時に自動的
終了	ユーザ操作	システム終了時に自動的
GUI	あり	なし
ネットワーク対応	基本的にはなし	あり
同時操作	基本的にはなし	あり
メリット	実装が簡単 動作テストが直感的 単体で使用可能	常時起動 ネットワーク対応による拡張性 複数人での同時使用可能
デメリット	起動・終了時に人為的ミスの可能性 ネットワーク対応の恩恵なし 複数人での同時使用不可	実装が複雑 動作テストが直感的でない GUI を別途用意する必要あり

4.2 デーモンのユーザインターフェース

デーモンソフトウェアは GUI を持たないので、ソフトウェアを操作するためには何らかのインターフェースが必要である。ftpd や httpd など良く知られているデーモンソフトウェアは、ネットワークを介してファイルに関する機能を提供するため、インターフェースはソケットを使ったネットワーク通信となっている。artsatd も地上局という機能をネットワークを介して提供していると考えられるため、インターフェースはソケットを使ったネットワーク通信となる。

次に、artsatd とユーザの間でどのようなプロトコルの通信を行うかを検討する。地上局ソフトウェアを使用するユーザは、最終的には GUI として表示されることを望んでいることがほとんどであるので、artsatd が HTML で記述された簡易的な GUI をユーザに送信し、ユーザはウェブブラウザ^{*13}を使ってこれを表示すると都合が良い【図 7】。この時、artsatd は見かけ上ウェブサーバとなり、ユーザからの“HTTP GET”リクエストに“HTTP OK”を返し、GUI となるコンテンツを送信する。

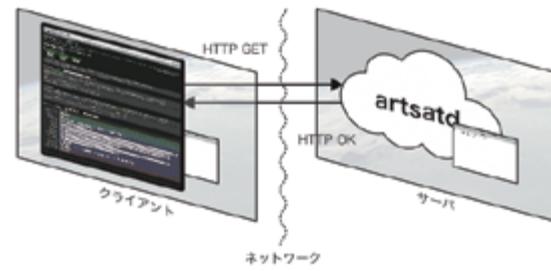


図 7 HTML で記述された GUI

また、GUI を介さずに地上局のすべての機能を実行できると、外部の各種ネットワークサービスと連携することが可能となる。REST^{*14} や SOAP^{*15} などが実現手段として候補に挙がるが、artsatd では動作が軽く実装が容易な JSONRPC^{*16} を採用する。JSONRPC も HTTP 上に実装できる²⁾ため、HTML を利用した GUI とも相性が良い。

JSONRPC 形式のインターフェースでは、地上局のすべての操作に対応するメソッドを提供する。メソッドを system・observer・database という 3 つのカテゴリに分類し、それぞれ、汎用的な処理・地上局に関する処理・衛星のデータベースに関する処理を表すこととする。すべてのメソッドの一覧を【表 3】に示す。

これらの JSONRPC のメソッドを使用することによって、HTML を利用した GUI だけでなく、高度な表現を備えた専用の GUI アプリケーションを独立して開発することも可能となる。

artsatd では【表 4】のように HTML 形式のインターフェースをポート 16780 (ART80) 番で、JSONRPC 形式のインターフェースはポート 16782 (ART82) 番で通信するものとする。

^{*13} Safari 8 での動作確認を行なっている

^{*14} Representational State Transfer - http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

^{*15} Simple Object Access Protocol - <http://www.w3.org/TR/soap12-part0>

^{*16} JavaScript Object Notation Remote Procedure Call - <http://www.jsonrpc.org>

表 3 提供される JSONRPC のメソッド

メソッド名	動作内容
system.rpcEcho	エコーを返す
observer.getVersion	artsatd のバージョンを取得
observer.getSession	セッションを取得
observer.setManualRotator	ローテータ手動モードの状態を設定
observer.getManualRotator	ローテータ手動モードの状態を取得
observer.setManualTransceiver	無線機手動モードの状態を設定
observer.getManualTransceiver	無線機手動モードの状態を取得
observer.setManualTNC	TNC 手動モードの状態を設定
observer.getManualTNC	TNC 手動モードの状態を取得
observer.setEXNORAD	追尾する衛星の拡張 NORAD 番号を設定
observer.getEXNORAD	追尾する衛星の拡張 NORAD 番号を取得
observer.setMode	地上局の動作モードを設定
observer.getMode	地上局の動作モードを取得
observer.getTime	内部時刻を取得
observer.getObserverCallsign	地上局のコールサインを取得
observer.getObserverPosition	地上局の位置を取得
observer.getObserverDirection	地上局のアンテナの方位角と仰角を取得
observer.getObserverFrequency	地上局の無線機の送受信周波数を取得
observer.getSpacecraftPosition	衛星の位置を取得
observer.getSpacecraftDirection	衛星の方位角と仰角を取得
observer.getSpacecraftDistance	衛星までの距離を取得
observer.getSpacecraftSpeed	衛星の速度を取得
observer.getSpacecraftFrequency	衛星と通信する時の送受信周波数を取得
observer.getSpacecraftDopplerShift	衛星と通信する時のドップラーシフト率を取得
observer.getSpacecraftAOSLOS	衛星の AOS と LOS を取得
observer.getSpacecraftMEL	衛星の最大仰角を取得
observer.getRotatorStart	ローテータの制御開始時刻を取得
observer.getError	内部エラーを取得
observer.isValidRotator	ローテータが正常に動作しているかを判定
observer.isValidTransceiver	無線機が正常に動作しているかを判定
observer.isValidTNC	TNC が正常に動作しているかを判定
observer.controlSession	操作権限を取得
observer.excludeSession	排他的な使用を申請
observer.setRotatorAzimuth	ローテータの方位角を設定
observer.setRotatorElevation	ローテータの仰角を設定
observer.setTransceiverMode	無線機モードを設定
observer.setTransceiverSender	無線機の送信周波数を設定
observer.setTransceiverReceiver	無線機の受信周波数を設定
observer.setTNCMode	TNC のモードを設定
observer.sendTNCPacket	TNC にデータを送信
observer.requestCommand	衛星に向けた制御コマンドの送信を要請

メソッド名	動作内容
database.setName	衛星の名前を設定
database.getName	衛星の名前を取得
database.setCallsign	衛星のコールサインを設定
database.getCallsign	衛星のコールサインを取得
database.setRadioBeacon	衛星のビーコン情報を設定
database.getRadioBeacon	衛星のビーコン情報を取得
database.setRadioSender	衛星のコマンド送信情報を設定
database.getRadioSender	衛星のコマンド送信情報を取得
database.setRadioReceiver	衛星のコマンド受信情報を設定
database.getRadioReceiver	衛星のコマンド受信情報を取得
database.setOrbitData	衛星の軌道要素を設定
database.getOrbitData	衛星の軌道要素を取得
database.getCount	登録されている衛星の数を取得
database.getField	登録されている衛星の情報を取得
database.getFieldByName	名前で衛星の情報を検索
database.getFieldByCallsign	コールサインで衛星の情報を検索
database.getEXNORADByName	名前で衛星の拡張 NORAD 番号を検索
database.getEXNORADByCallsign	コールサインで衛星の拡張 NORAD 番号を検索

表 4 インターフェースとプロトコル

インターフェース	プロトコル	ポート
GUI	HTML over HTTP 1.0	16780
RPC	JSONRPC 2.0 over HTTP 1.0	16782

4.3 同時アクセスと操作権限

ネットワークを介してユーザが地上局ソフトウェアを操作する場合、ユーザの操作が遠隔地でいつ発生するかを予想することはできず、操作を行うユーザが1人なのか複数人なのかも予想することはできない。そこで、必然的に複数のユーザの同時アクセスに対応しなければいけなくなる。

地上局のハードウェアリソースは1つであるため、複数のユーザが同時に非同期な操作を行うことは予測できない不具合を引き起こす。ロック機構を使い非同期なアクセスが発生しないように制限することでこの問題を解決することができる³⁾。例えば、あるユーザが無線機の制御コマンドをシリアルポートを介して送信中に、別のユーザが非同期に無線機の操作を行うとき、1つしかないシリアルポート上でコマンドが混じり合ってしまう正しく制御を行うことができなくなる。シリアルポートにロック機構を設けることで非同期な書き込みは回避できるが、コマンドのシーケンスレベルでは回避することが難しい。コマンドのシーケンスを1つの単位としてロック機構を設けると、コマンドが混じり合う不具合を解決することはできる。しかしながら、衛星の追尾操作という複数のハードウェアを連携させる一連の流れを持った操作の中では、さらにまとまった単位でのロック機構の導入が必要となる。

そこで artsatd では、システムに1つだけ操作権限を導入することとする。地上局の操作を望むユーザは操作権限を取得することによって、操作権限を保持しているあいだ地上局を独占的に操作することが可能となる。操作権限はユーザレベルでのロック機構と考えることができる。操作権限の排他的な使用を申請しない場合には、他のユーザの操作権限の取得によって、権限が失われることがある。他のユーザに操作権限を譲渡したくない場合には、操作権限の排他的な使用を申請する。ユーザが操作を行なっているあいだ操作権限の有効期限を延長し、ユーザが操作を終了したり、ネットワークの不調などなんらかの原因で操作を継続することが不可能になった場合には、有効期限が切れた操作権限を

破棄する。操作権限を保持していないユーザは、地上局の監視のみ行うことができる。操作権限に関する状態の遷移を【図8】に示す。

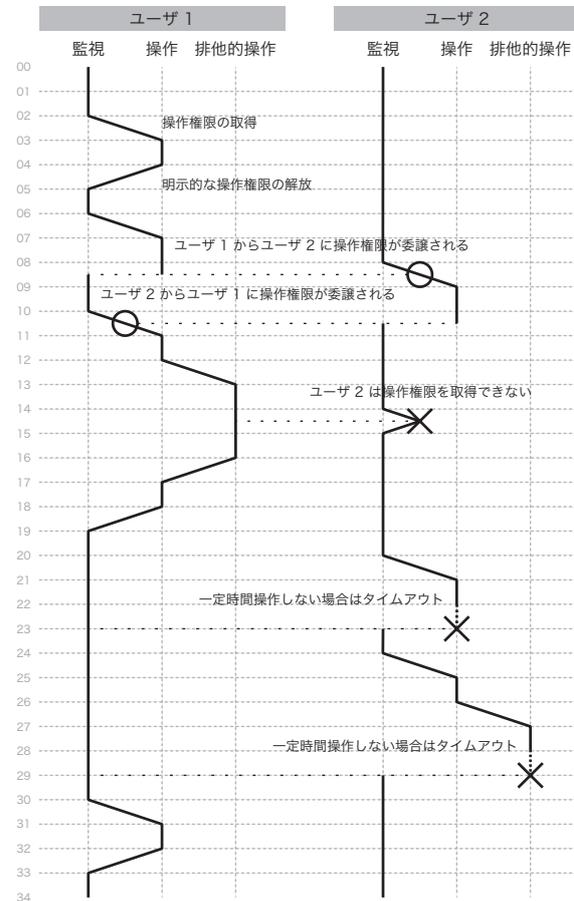


図8 操作権限に関する状態の遷移

4.4 セキュリティ対策

不特定多数のユーザが遠隔地から地上局ソフトウェアにアクセスできる状況は、セキュリティの観点からは都合が悪い。特に地上局を操作することに関しては電波法との兼ね合いに注意する必要がある。免許を持った管理者が不在の状況で遠隔地から地上局を操作して電波を送信してはならない。地上局の監視のみ行う場合であっても、悪意を持った多数のアクセスにより DoS 攻撃と同じ状況となり、地上局のシステムがダウンする可能性がある。

そこで、地上局の操作をローカルネットワークからのみに制限する。ローカルネットワークとは localhost^{*17} を表し、IP アドレスは 127.0.0.1 である。ローカルネットワーク以外の IP アドレ

*17 自機 IP アドレスとサブネットマスクに関する処理を追加すると、firewall 内部のアクセスを許可することもできる

スからのアクセスでは地上局の監視のみを行うことができるようにする。また、最大同時接続数を設定することにより、過大な数の同時接続からある程度システムを防御する。

遠隔地から地上局を操作したい場合には【コード1】のようにSSHポートフォワーディングを使用してローカルネットワークにトンネルを作成し、artsatdにアクセスする。artsatdはユーザアカウントとパスワードの管理を行わず、SSHに依存する設計とする。SSHのユーザアカウントを持った利用者が地上局の管理者の一員であることを前提条件とし、免許を持った管理者が地上局に所在することを確認した上で遠隔地から操作を行うこととなる。

コード1 SSHを使用したアクセス

```
ssh -l <user> -L 16780:localhost:16780
      artsat.idd.tamabi.ac.jp
```

このようなアクセス制限を行なってもソケットは依然として開いており、DoS攻撃の対象となる危険性があるので、重要なミッションではポート番号を変更したり外部にfirewallを設けるなど工夫が必要である。

4.5 衛星一覧と軌道要素の自動取得

衛星の一覧と軌道要素は、指定されたウェブサイトから自動的に取得されると便利である。専用に記述されたウェブサイトだけでなく、CelesTrakなど広く利用されている既存のウェブサイトをクリックできると効率的である。

ウェブサイトのフォーマットに合わせた解析クラスを用意し、フォーマットが同じであれば別のURLからもデータを取得できるようにする。このような設計により、新しいウェブサイトへの対応を迅速に行うことが可能となる。また、取得元のサイトを複数指定できるようにし、データに重複がある場合には元期の新しい軌道要素を優先することとする。収集された衛星の一覧と軌道要素はsqlite3形式のデータベースに格納する。

衛星のデータベースを管理したり検索したりするときには、一意の番号や名前を利用できると便利である。地球を周回する衛星について

はNORAD番号が割り当てられており、これは重複がない番号であるのでそのまま利用することができる。地球周回軌道ではない宇宙機や何らかの理由でNORAD番号を持たない機体のために、NORAD番号と共存できる新しい番号として、拡張NORAD番号を定義する。拡張NORAD番号は、100000未満の番号は通常のNORAD番号として扱い、そうでない場合は独自に定義された番号として扱うこととする。

4.6 衛星の軌道計算と監視

ユーザは追尾する衛星を拡張NORAD番号や衛星の名前で指定することができる。データベースに登録されている衛星の一覧から、拡張NORAD番号の一致する衛星や名前の一部が一致する衛星を検索して、追尾の対象に設定する。

追尾の対象に設定されている衛星の現在位置と移動方向は、SGP4 / SDP4 アルゴリズムやSGP8 / SDP8 アルゴリズムを使用して、TLEと現在時刻から計算する⁴⁾。衛星の軌道の計算にはcppOrbitTools^{*18}と呼ばれるライブラリを使用し、アルゴリズムはSGP4 / SDP4を用いる。次に、求めた衛星の現在位置と移動方向、さらに地上局の位置関係からドップラーシフトの割合を計算する。

追尾する衛星が指定されているあいだ軌道の計算を常に行い、UNIX時間を基準時間とし、時間の最小単位が1秒であるので1秒ごとに値を更新する。地上局と衛星の位置関係から衛星の方位角と仰角を計算し、仰角が0度以上になる場合を監視する。

UNIX時間はPOSIXにおいては閏秒が考慮されていない。秒速数キロという速度で飛ぶ衛星では1秒の差が大きな位置の差となって現れるが、地上から見た時の方位角と仰角の誤差は、アマチュアが利用可能なアンテナの最小駆動角度よりも十分に小さいことが想定され、アンテナの利得を考慮に入れても実用に耐える。さらに、時刻の情報は直近の軌道要素の原期との差を計算することに利用されるため、実質的な誤差はほとんど発生しないと考えられる。

artsatdではUNIX時間のPOSIX運用を前提

*18 <http://www.zeptomoby.com/satellites>

とし、TZ 環境変数^{*19}への閏秒補正^{*20}を設定せず、適宜 NTP^{*21}を使って閏秒補正後の時刻と同期する方式を採用する。

4.7 アンテナと無線機の制御

追尾する対象の衛星が可視範囲に存在するとき、アンテナの方位角と仰角、ドップラースhiftを含めた無線機の送受信周波数を連続的に制御する。

アンテナの制御では、実際にはアンテナに取り付けられたローテータを使用して方向を変えることになるが、ローテータが回転するには数秒から数十秒の時間が必要である。角度の差が大きいほど必要な時間は長くなり、特に0度と360度をまたぐ時に最も時間が長くなる。アンテナの回転が終了する前に次の角度を連続して指定するとローテータを破壊する危険性があるので、角度の指定には一定時間の間隔を設け、アンテナが指定された角度に向いてから次の角度を指定するようにする。

無線機の制御でも誤動作を防ぐために一定時間の間隔を設けて制御を行う。

衛星の仰角が0度以上となりアンテナの制御を始める場合、衛星の方位角がアンテナの方位角と大幅に異なると衛星を補足するまでに時間がかかる。そこで、次に衛星が出現する方位角をあらかじめ計算しておき、前もってアンテナの方位角を予想される角度に設定する。

4.8 音声信号の自動録音

衛星が可視範囲に存在し、衛星からの電波を受信できると考えられる場合には、無線機で受信された音声信号を自動的に録音する。artsatdが動作しているパーソナルコンピュータに無線機から音声信号を入力し、SoX^{*22}などのオーディオ編集ソフトウェアを使って録音する。

artsatdはソフトウェアの移植性を高めるためにPOSIX / BSD 準拠のソフトウェアとして互換性を保ちたいので、音声を扱うハードウェア固有のAPIを直接使用しないこととする。そこで、音声に関する処理を外部のソフトウェアに

委託する。

録音を開始する時点でSoXなどをfork-execしてartsatdの子プロセスとして実行し、録音の終了時に子プロセスを終了させる。別の手法として、artsatdの外部で動作するスクリプトを作成し、JSONRPCを介してartsatdと連携させ、SoXなど任意のソフトウェアを起動させたり終了させたりすることもできる。

4.9 衛星とのコマンドの送受信

ウェブブラウザで表示したGUIから制御コマンドを送信する場合やJSONRPCを利用する場合、短時間に多数のコマンドが送信されないように制限を行う必要がある。

制御コマンドはある程度の長さを持ったシーケンスであり、アマチュア無線帯の電波としてすべてを送信するには数秒程度の時間がかかる。送信されたコマンドを衛星が処理して応答を返すにも幾らかの時間が必要であり、複数のコマンドを一度に送信すると衛星にとって致命的な不具合を引き起こす一因となる。

そこで、ユーザが制御コマンドを送信するときには送信されるコマンドを内部キューに一旦登録し、一定時間の間隔で送信されるように調節を行う。衛星から応答コマンドを受信した場合には、それを表示してデータベースに記録する。

4.10 ウェブカメラと音声信号のストリーミング

遠隔地からアンテナの状態を監視するためにはウェブカメラを使用する。また、無線機で受信された音声を遠隔地で聴くために、インターネットラジオのサーバソフトウェアを使用したストリーミング配信を行う。どちらもウェブブラウザにURLを入力することでアクセスできる。

artsatdのGUIはHTMLで記述され、ユーザはウェブブラウザを利用してartsatdを操作するため、HTMLの内容を工夫することでウェブブラウザ上でウェブカメラやストリーミング配信と連携させることができる。

^{*19} タイムゾーンやサマータイム、閏秒などの設定を指定する環境変数

^{*20} TZ=right/Asia/Tokyo など

^{*21} Network Time Protocol - <http://www.ntp.org>

^{*22} Sound eXchange - <http://sox.sourceforge.net>

4.11 仕様のまとめ

上記の検討から artsatd の仕様を以下にまとめる。

- POSIX / BSD 準拠の関数やライブラリを用いる
- システムの起動時に自動的に起動し常駐する
- グラフィカルユーザインターフェース (GUI) を持たず、ネットワークからのアクセスにより操作する
- ネットワークからのアクセスにいつでも応答し、複数のユーザの同時アクセスにも対応する
- 複数のユーザが同時に地上局を操作しようとする場合、操作権限は 1 人だけに与えられるものとする
- 操作権限はユーザ間で譲渡することが可能であり、必要があれば排他的に使用することもできるものとする
- 地上局の操作はローカルネットワークからのアクセスのみ可能とし、リモートネットワークからは監視のみ可能とする
- リモートネットワークから操作する場合には、セキュリティを考慮し、SSH ポートフォワードリングを使用してアクセスする⁵⁾
- 衛星の一覧や軌道要素の情報は、指定されたウェブサイトをクリックして、最新のデータを自動的に収集する
- 追尾する衛星を 1 つ選択することができる
- 追尾している衛星の軌道を常時計算し、衛星の到来を監視する
- 衛星が可視範囲に存在するとき、衛星に向けてアンテナを制御し、ドップラーシフトを考慮して無線機の周波数を設定する
- 衛星が可視範囲に存在するとき、受信された音声信号を自動的に録音して蓄積する
- 衛星に固有の制御コマンドを衛星に向けて送信することができる
- 衛星から送信された応答コマンドを受信することができる
- ウェブカメラと連携してアンテナの状態を監視することができる
- 無線機の音声信号をストリーミングすることができる

5 実装の手法

5.1 開発言語の選択

ネットワークに対応した地上局ソフトウェアの開発では、高レベルなサーバ・クライアント型の通信処理の構築から、低レベルなシリアル通信を用いた地上局のハードウェアの制御まで、幅広い階層にまたがる処理を実装する必要がある。特に、システムに常駐するためにはプロセスのデーモン化^{*23}を行うことが大切であり、常駐して軌道を計算するためには高速な処理能力が要求される。エラー発生時に確実に復帰して軌道計算を続行するため、システムコールに対するエラー処理の実装やシグナルハンドラの実装など、エラー処理に関わる低レベルな実装を記述できる必要もある。

これらの要件から、C 言語・C++ 言語・Objective C 言語などが候補に上がるが、使用しているライブラリとの関係性や他の環境への移植性を考慮して、C++ 言語を採用する。

5.2 artsatd の開発環境

C++ 言語を使用する場合、サーバ・クライアント型の通信処理を一から構築すると非常に手間がかかり実装も複雑になる。そこで、cpp-netlib^{*24} と呼ばれる、テンプレートを駆使して実装された高機能なネットワークライブラリを使用する。cpp-netlib は boost ライブラリ^{*25}を同時に必要とするため、boost ライブラリも使用することになる。正規表現などの高度な文字列処理の煩雑さは、boost ライブラリの使用により解決される。

次に、artsatd は POSIX / BSD 互換のソフトウェアを目指して開発されているため、ハードウェアや OS 固有の API を使用せず、原則として POSIX で提供されている API のみを用いて実装を行う。artsatd がソースコードレベルで使用するライブラリについても、POSIX に準拠したライブラリを採用する。

artsatd は【表 5】の開発環境を用いて開発されたが、容易に他の環境に移植できることが期待される。

^{*23} fork(2) 関数を使用して、プロセスをバックグラウンドに移行し、標準入出力を閉じるなどの処理を行う

^{*24} <http://cpp-netlib.org>

^{*25} <http://www.boost.org>

5.5 ハードウェア制御の実装

アンテナローテータ・無線機・TNC は RS-232C 規格のシリアルポートを使った接続となっており、artsatd の開発環境である Apple iMac^{*29} とは USB シリアル変換器^{*30} を使って接続する。このような変換器は、ソフトウェア側からはシリアルポートとして扱うことが可能である。

artsatd では、ハードウェア関連のクラスとして、汎用的なシリアル通信を行うための抽象クラス TGSDeviceInterface を定義し、シリアルポートのオープン・クローズ処理や多バイト長のデータの送受信などの基本的な機能を提供する。さらに、TGSDeviceInterface クラスを継承したクラスとして、汎用的なアンテナローテータの機能を提供する抽象クラス TGSRotatorInterface、汎用的な無線機の機能を提供する抽象クラス TGSTransceiverInterface や汎用的な TNC の機能を提供する抽象クラス TGSTNCInterface を定義している。これらの抽象クラスを継承して、具体的なハードウェアに対応する機種依存の実装を行う。

“TGS” のプレフィックスを持つハードウェア関連のクラスは、リエントラントではないため、マルチスレッドの環境下での使用には注意を要する。artsatd では多数のスレッドを同時に動かすため、ハードウェア関連のクラスのインスタンスにも非同期にアクセスが行われることから、ロック機構を使った同期化が必要となる。このため、ハードウェアへのアクセスのスケジューリングと同期化を行うための抽象クラス ASDDeviceInterface を定義し、具体的なハードウェアごとにこのクラスを継承した具象クラスを実装している。ASDDeviceInterface クラスは、boost::unique_lock<boost::mutex> を用いて単純なロック機構を構成している。

5.6 ネットワーク対応の実装

5.6.1 cpp-netlib の利用

artsatd では、サーバ・クライアント型の通信処理を実装するために cpp-netlib を利用する。cpp-netlib を使用すると、ウェブサイトへのアクセスやウェブサーバ機能の実装に必要な煩雑な通信処

理部分を実装することなく、アプリケーションに固有の処理のみを実装することが可能となる。

クライアント側の実装では boost::network::http::client を利用し、サーバ側の実装では boost::network::http::server を利用する。

5.6.2 ウェブサイトのクローラ

ウェブサイトをクリックする基本的な機能を提供するクラスとして、抽象クラス ASDHTTPClientInterface を定義する。ASDHTTPClientInterface クラスは、スレッドを 1 つ生成し、その上で boost::network::http::client のインスタンスを使用してウェブサイトへの接続を行う。ASDHTTPClientInterface クラスを継承した、CelesTrak や他のウェブサイトをクローラする具象クラスは、ウェブサイトの具体的なフォーマットを解析する機能のみを実装する。

設定された時間ごとにウェブサイトに接続して情報の更新を監視し、軌道要素の情報を抽出できた場合には、衛星の NORAD 番号・名前・軌道要素を sqlite3 形式のデータベースに格納する。

5.6.3 サーバとしての動作

HTML サーバと JSONRPC サーバに共通する基本的な機能を提供するクラスとして、具象クラス ASDNetworkServer を定義する。このクラスは、HTTP 接続に関するリクエストヘッダの抽出やリクエストボディの取得、レスポンスヘッダの設定やコンテンツの設定、クッキーの処理などを行う。

HTML サーバに固有の処理は ASDServerOperation クラスに実装されており、ASDServerOperation クラスのインスタンスを ASDNetworkServer クラスのインスタンスに登録する形式で動作を行う。

ASDServerOperation クラスは、ユーザアクセス時の GET パラメータの内容により地上局の操作を行うように設計されている。ユーザに送信される HTML は、動的に生成される部分とテンプレートファイルから取得される部分が融合したものとなる。テンプレートファイルは、動的に生成される部分を “<![TAG />” や “<![TAG />” ~ “<![TAG />” の特殊タグを使って記述した

^{*29} Apple iMac 27-inch, Late 2009 / 2.8GHz Intel Core i7, 4GB 1067MHz DDR3 / OS X 10.9.5 (13F1077)

^{*30} RATOC Systems, Inc. 製 REX-USB60F と ELECOM CO., LTD. 製 UC-SGT での動作確認を行なっている

変則的 HTML ファイルである【コード 4】. 特殊タグの名称と置換される具体的な内容の対応は, ASDServerOperation クラスのソースコードに記述されている.

コード 4 テンプレートファイル内の 特殊タグ (一部抜粋)

```
<div id="status">
  <div>

    <!-- <!TAG /> は, 数値や文字列に置換される -->
    <a href="?shrink=status">[ <pre><!ST /></pre>
      ]</a> <b>Spacecraft Status:</b>
  </div>

  <!-- <!TAG /> ~ <!TAG /> は, 特殊タグに囲まれた
  範囲が別のタグ構造に置換される -->
  <![STT />
  <iframe class="orbital"
    src="orbital.html" seamless></iframe>
  <div class="space"></div>
  <![STT />
</div>
```

JSONRPC サーバに固有の処理は ASDServerRPC クラスに実装されており, HTML サーバの場合と同じく, ASDServerRPC クラスのインスタンスを ASDNetworkServer クラスのインスタンスに登録して動作を行う.

ASDServerRPC クラスは, ユーザアクセス時のリクエストボディを取得して JSONRPC の関数呼び出しのパラメータを解析し, 対応する artsatd 内部の関数を呼び出したのち, JSONRPC の戻り値として結果を送信する.

JSONRPC 2.0 仕様のフォーマット^{*31}に対応し, Notification^{*32} や Batch^{*33} にも対応している. 【コード 5】に JSONRPC の関数呼び出しと戻り値の例を示す.

コード 5 JSONRPC の関数呼び出しと戻り値

```
// http://<domain>:16782/rpc.json への関数呼び出し
{
  "id" : 1,
  "jsonrpc" : "2.0",
  "method" : "observer.getSpacecraftPosition",
  "params" : {
    "session" : "5588b9278ab61961"
  }
}
```

```
// 戻り値
{
  "id" : 1,
  "jsonrpc" : "2.0",
  "result" : {
    "altitude" : 403.703104907,
    "latitude" : 45.261216507,
    "longitude" : -114.071108046,
    "session" : "5588b9278ab61961"
  }
}
```

5.7 非同期な同時アクセスへの対応

システムに 1 つしか存在しないハードウェアやデータベースに, 任意のスレッドから非同期にアクセスするためには, ロック機構を用いた同期化が必要となる. lock / unlock を用いて単純に同期化を行うことで目的は達成されるが, 読み出し動作のみの同時アクセスであってもロックを行うため効率が悪い.

そこで artsatd では, 効率的なロックを行うために boost::shared_lock<boost::shared_mutex>, boost::upgrade_lock<boost::shared_mutex>, boost::upgrade_to_unique_lock<boost::shared_mutex> を使用する. これらのロックは共有ロックと呼ばれており, 読み出し動作同士の間にはロックを行わず, 読み出し動作と書き込み動作, 書き込み動作同士の間にはロックを行うため効率が良い⁷⁾. artsatd では, ハードウェアやデータベースに対しての書き込み動作よりも読み出し動作の方が圧倒的に多いため, 共有ロックを使用するとパフォーマンスを大幅に向上させることができる.

同期化したい 1 つのリソースに対して, 1 つの boost::shared_mutex のインスタンスを割り当てる設計とし, 上記 3 つのクラスを【コード 6】に示すように厳格に使用した実装を行う.

コード 6 共有ロックを用いた効率化(一部抜粋)

```
tgs::TGSError artsatd::setEXNORAD(
  std::string const& session, int exnorad)
{
  if (!session.empty()) {
    // _mutex_session を共有ロック
    boost::shared_lock<boost::shared_mutex>
      rlock(_mutex_session);
```

^{*31} <http://www.jsonrpc.org/specification>

^{*32} 戻り値を必要としない関数呼び出し

^{*33} 複数の関数呼び出しと戻り値の取得をまとめて行う呼び出し

```

    if (_session.owner == session) {

        // _mutex_control を書き込みロック
        boost::unique_lock<boost::shared_mutex>
            wlock(_mutex_control);
        _control.exnorad =
            (exnorad >= 0) ? (exnorad) : (-1);
    }
}
return error;
}

int artsatd::getEXNORAD(void) const
{
    // _mutex_control を共有ロック
    boost::shared_lock<boost::shared_mutex>
        rlock(_mutex_control);
    return _control.exnorad;
}

tgs::TGSError artsatd::excludeSession(
    std::string const& session, bool exclusive)
{
    if (!session.empty()) {

        // _mutex_session を書き込み予約ロック
        boost::upgrade_lock<boost::shared_mutex>
            ulock(_mutex_session);
        if (_session.owner == session) {

            // _mutex_session を書き込みロック
            boost::upgrade_to_unique_lock<
                boost::shared_mutex> wlock(ulock);
            _session.exclusive = exclusive;
        }
    }
    return error;
}
}

```

5.8 sqlite3 データベースの操作

衛星一覧や軌道要素の情報は、sqlite3 形式のデータベースに格納される。データベースはシステムに 1 つだけ存在し、データベースへのアクセスは複数のスレッドから非同期に行われる。そこで、データベースの操作に関してもロック機構を使った同期化を行う必要があるが、sqlite3 はデータベース自身が複数スレッドからのアクセスに対応しており、適切なコンパイルオプションの選択と、1 つのスレッドに 1 つのデータベースコネクションを作成することで対応することができる。

データベースの操作に関する、オープン・クローズ・作成・追加・更新・削除・検索などの基本的な機能を提供するクラスとして、抽象クラス TGSDatabaseInterface を定義し、

TGSDatabaseInterface クラスを継承した具象クラス内で内容に応じた具体的な処理を実装する。

効率的な処理を行うために、連続した追加や更新を行う場合にはトランザクションを使用し、“BEGIN TRANSACTION;” から “END TRANSACTION;” までの間でデータベースの操作を行う。また、データベースを閉じる前に “VACUUM;” を実行して、不要となった空き領域を解放しておくが良い。TGSDatabaseInterface クラスはこれらを実行する関数として、begin() 関数・end() 関数・vacuum() 関数を提供している。

5.9 衛星の軌道計算の実装

衛星の軌道計算に関する共通した機能を提供するクラスとして、抽象クラス TGSOrbitInterface を定義し、軌道計算のアルゴリズムに応じて TGSOrbitInterface クラスを継承した具象クラスを実装する。

TLE から SGP4 / SDP4 アルゴリズムを使用して軌道を計算するクラスは、内部で cppOrbitTools を使用した実装を行う。軌道計算の処理速度を向上させるために、cppOrbitTools が必要とするいくつかのクラスのインスタンスをキャッシュし、不必要な生成と解放を防止する。

cppOrbitTools は、衛星の現在位置と移動方向を計算することができるが、無線周波数のドップラーシフトの割合を計算することはできない。そこで、地上局の位置と衛星の現在位置や移動方向から、ドップラーシフトの割合を独自に計算する。

5.10 JSONRPC を利用した自動録音

音声信号の自動録音は、artsatd が SoX を子プロセスとして起動して行うこともできるが、JSONRPC を利用して外部のスクリプトとして動作させることもできる。

この方法は、生成された音声ファイルをファイル共有サービスやウェブサイト自動的に転送する場合に特に有効であり、artsatd のソースコードを修正せずにさまざまな機能拡張を行うことができる。【コード 7】に衛星が可視範囲に入ったら SoX を起動して録音を行い、可視範囲から出た時に SoX を終了し、録音されたファイルを dropbox^{*34} にアップロードする ruby スクリプトを示す。

^{*34} <http://www.dropbox.com>

コード 7 自動録音のための ruby スクリプト (一部抜粋)

```

threshold = -1.0
$session = ""

def get_elevation()
  begin
    http = Net::HTTP.new('localhost', 16782)
    req = JSON.generate({
      "jsonrpc" => "2.0",
      "id" => 0,
      "method" =>
        "observer.getSpacecraftDirection",
      "params" => {"session" => $session})
    response = JSON.parse(
      http.post('/rpc.json', req).body)
    $session = response["result"]["session"]
    response["result"]["elevation"]
  rescue Timeout::Error
    -360
  rescue
    -360
  end
end

def get_visible()
  begin
    http = Net::HTTP.new('localhost', 16782)
    req = JSON.generate({
      "jsonrpc" => "2.0",
      "id" => 0,
      "method" =>
        "observer.getSpacecraftAOSLOS",
      "params" => {"session" => $session})
    response = JSON.parse(
      http.post('/rpc.json', req).body)
    $session = response["result"]["session"]
    aos = Time.parse(response["result"]["aos"])
    los = Time.parse(response["result"]["los"])
    (aos <= Time.now && Time.now <= los)
  rescue Timeout::Error
    nil
  rescue
    nil
  end
end

def get_dbox_link(path)
  begin
    access_token = 'your token'
    client = DropboxClient.new(access_token)
    client.shares(path)["url"]
  rescue
    ""
  end
end

loop do
  e = get_elevation()
  v = get_visible()
  puts "elevation: #{e}"
  if (v)
    now = Time.now
    time = now.strftime("%Y%m%d_%H.%M")
    basename = "/Users/artsat/Dropbox
      /autorec/INVADER_#{time}.ogg"
    dbox_name = "/autorec/INVADER_#{time}.ogg"
    puts "● start recording... > #{basename}"
    systemu "rec #{basename} channels 1" do

```

```

|pid|
  count = 0
  while (v || (count < 24 && v == nil))
    if v == nil
      count += 1
    else
      count == 0
    end
    e = get_elevation()
    v = get_visible()
    puts "● elevation: #{e}"
    sleep 5
  end
  Process.kill :INT, pid
end
puts "■ stop recording..."
link = get_dbox_link(dbox_name)
if link.size > 0
  puts "public link -> #{link}"
end
end
sleep 2
end

```

6 運用の実証と成果

6.1 INVADER 衛星の運用

6.1.1 衛星の概要

INVADER 衛星は、ARTSAT プロジェクトが開発した宇宙機の 1 号機として 2014/02/28 JST に打ち上げられた、1UCubeSAT サイズの超小型衛星である。186 日間に渡り高度 400km 付近を飛行し、2014/09/02 JST に大気圏に再突入した。【表 6】に INVADER 衛星の仕様を【図 10】に実機の写真を示す。

表 6 INVADER 衛星の仕様

項目	値
NORAD 番号	39577
コールサイン	JQ1ZKK
オスカーナンバー	CO-77
投入軌道	低軌道 400km
アンテナ形状 (downlink)	1/2 波長ダイポール
アンテナ形状 (uplink)	1/2 波長モノポール
CW 出力	100mW
CW 周波数	437.325MHz
CW 変調方式	無変調 + モーリス符号
FM 出力	800mW
FM 周波数 (downlink)	437.200MHz
FM 周波数 (uplink)	145MHz 帯
FM 変調方式	FSK + AX.25 1200bps

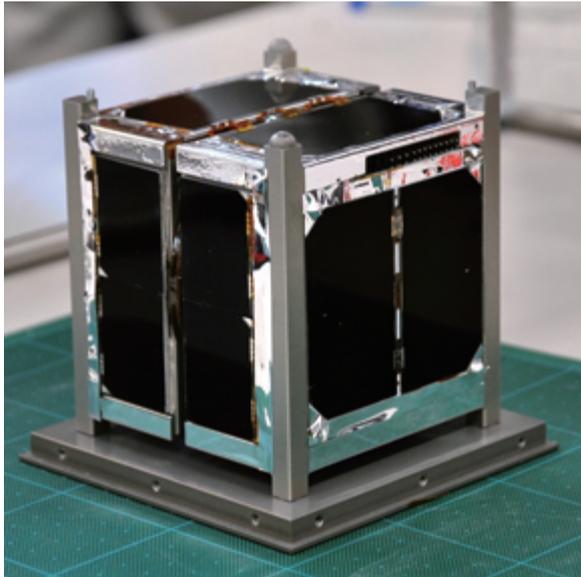


図 10 INVADER 衛星の実機

6.1.2 運用結果と課題

INVADER 衛星は、打ち上げ当初から多摩美地上局での CW 信号の受信に成功し、artsatd の軌道計算結果はほぼ正しい値を示していた。artsatd の自動追尾機能と遠隔地からの監視や操作の機能を活用して、すべての設定されたミッションに成功し、地球の写真を撮影することにも成功した⁸⁾。

しかしながら INVADER 衛星の運用では、衛星からの応答コマンドのデコード成功率が低いことが度々問題になり、写真の撮影では何度も同じフレームをダウンリンクする必要があるなど運用が難しい点も存在した。

デコードに失敗する原因としては、衛星の電波送信機の問題、地上局付近の環境ノイズの問題や地上局のアンテナから無線機までの配線の問題などが検討されたが、国道 16 号線が付近に存在し、夜間はノイズが少なくなることなどから環境ノイズに影響されていた疑いはあると考えられている。また、アンテナから無線機までの配線が長く、S/N 比が低下しやすい環境であることも一因だと考えられる。

衛星の運用期間中に、デコード成功率の低さを補うための試行錯誤が繰り返され、ハードウェアモデムを使ったデコードだけではな

く、既存のソフトウェアモデム^{*35}の利用や INVADER 衛星に特化した独自のソフトウェアモデムの開発も試みられた【図 11】。独自に開発されたソフトウェアモデムは Invasive^{*36} と呼ばれ、JSON インターフェースを備えた設計となっている。

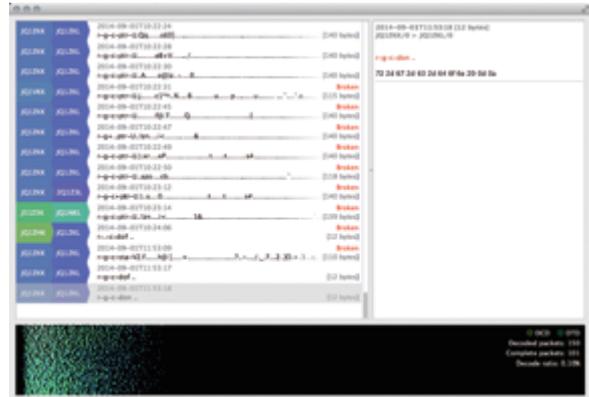


図 11 Invasive を使ったデコード

6.1.3 コマンドの自動送受信

INVADER 衛星の運用では、コマンドの送受信を自動化する実験が行われた。artsatd は、衛星に固有の制御コマンドを自動的に送信する機能を搭載しておらず、仮に自動的に送信するには、運用の内容に応じて、制御コマンドの送信と応答コマンドの受信を順序立てて管理する必要がある。

多摩美地上局の運用で使用された Invasive は、衛星からの応答コマンドのデコードに成功すると、JSON インターフェースを使用して結果を外部のソフトウェアに送信することができる。そこで、JSONRPC を使って Invasive と artsatd を連携させる ruby スクリプトを作成し、予定する運用でのコマンドの流れを実装した。このスクリプトを動作させることで、自動的にコマンドを送受信できることが期待されたが、コマンドの送受信の物理的な成功率が想定ほど良くなかったことと、送受信エラーが発生した場合のシーケンス制御が複雑になったことなどから実際の運用では利用されなかった。

^{*35} soundmodem - <http://uz7.ho.ua/packetradio.htm>

^{*36} <http://github.com/h2so5/Invasive>

6.2 DESPATCH 宇宙機の運用

6.2.1 宇宙機の概要

DESPATCH 宇宙機は、ARTSAT プロジェクトが開発した宇宙機の 2 号機として 2014/12/03 JST に打ち上げられた、50cm 立方サイズの小型宇宙機である。地球脱出軌道に投入され、2015/01/03 JST に運用を終了したのちも太陽の周りを周回し続けている。【表 7】に DESPATCH 宇宙機の仕様を【図 12】に実機の写真を示す。

表 7 DESPATCH 宇宙機の仕様

項目	値
NORAD 番号	40321
コールサイン	JQ1ZNN
オスカーナンバー	FO-81
投入軌道	地球脱出軌道
アンテナ形状 (downlink)	1/4 波長モノポール
アンテナ形状 (uplink)	パッチアンテナ
送信出力	7W
送信周波数	437.325MHz
送信変調方式	無変調 + モーリス符号・バーコード・PWM
受信周波数 (uplink)	145MHz 帯
受信変調方式	FSK

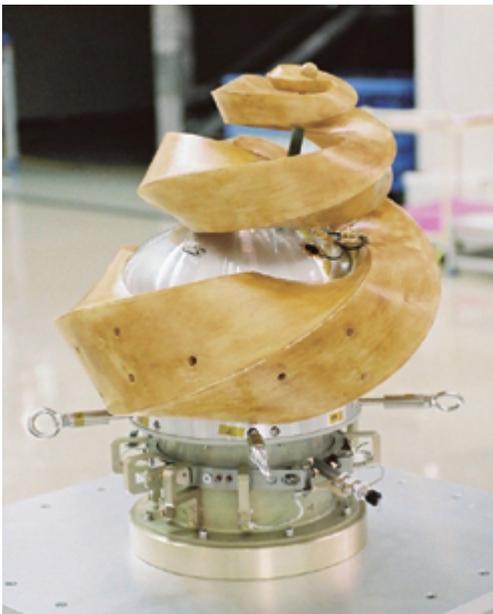


図 12 DESPATCH 宇宙機の実機 (写真 ©JAXA)

6.2.2 地球脱出軌道への対応

DESPATCH 宇宙機の運用では、衛星軌道ではなく地球脱出軌道への対応を行う必要があった。artsatd の開発当初は、cppOrbitTools を使用して計算できる地球周回軌道のみをサポートしていたが、DESPATCH 宇宙機の打ち上げに向けて機能の追加を行った。

初めに、DESPATCH 宇宙機が飛行する軌道を表現するために TLE に変わる新しいフォーマットを定義し、これを SCD^{*37} とした【コード 8】。SCD は、DESPATCH 宇宙機の打ち上げ時の初期パラメータと日時などを含む 3 行のフォーマットである。SCD の最新情報は、github のリポジトリ^{*38} で管理し、値が変更された場合にはリポジトリを更新する。

コード 8 地球脱出軌道を表す SCD

```

DESPATCH
SCD00001 56994.264023 10697006.0 1094554.0 // 改行なし
-1056032.0 5395.222 6937.623 -4193.223
150.0

SIN'EN2
SCD00002 56994.261129 9230208.0 -645302.0 // 改行なし
-161.0 6392.116 6952.874 -4240.850
150.0

```

DESPATCH 宇宙機には当初 NORAD 番号が割り当てられていなかったため、拡張 NORAD 番号を割り当ててシステムの開発を行った。そのため DESPATCH の SCD には、拡張 NORAD 番号が付されている。

次に、CelesTrak のウェブサイトではなく、github のリポジトリをクロールする新しいクラスを実装し、github のリポジトリからも軌道要素を取得できるように改良した。SCD で記述された軌道要素にも対応できるようにデータベースを拡張し、ユーザインターフェース周辺にも改良を加えた。

SCD を使用した軌道の計算は、DESPATCH プロジェクトで新しく開発された SCTracker^{*39} という積分型の軌道予測アルゴリズムを使用し、TLE の場合と同様に TGSOrbitInterface クラスを継承した具象クラスを実装した。

^{*37} SpaceCraft Description

^{*38} <http://github.com/ARTSAT/SCD>

^{*39} http://github.com/ARTSAT/ground_station/tree/master/common/libs/TGS/src/physics/despatch_tracker

これらの改良を行うことにより, artsatd を地球脱出軌道にも対応させ, DESPATCH 宇宙機の運用を行った。

6.2.3 運用結果と課題

DESPATCH 宇宙機の運用は, 打ち上げ当初は順調であった。2014/12/03 18:52 JST 前後の想定された時刻に, 初めての信号を受信することに成功し, 途中信号が弱く受信音を正確に聴くことができなくなったものの, 地上局の動作としてはほぼ想定された通りの動作を行った。

受信音が弱くなった原因は, 宇宙機の回転運動により宇宙機側のアンテナの向きが変わってしまったことによると考えられた。

第 1 フェーズのモールス信号を 19 時ごろから断続的に受信することに成功し, 23 時半ごろに信号を受信することができなくなった。第 1 フェーズから第 2 フェーズへ切り替わる時刻が 2014/12/04 00:20 JST に予定されており, DESPATCH 側のソフトウェアの不具合などが疑われたが, 2014/12/04 00:44 JST にはっきりとした第 2 フェーズのボーコード【図 13】を受信することに成功し, フェーズの移行が順調に成功したことが確認された。第 1 回目のパスの運用では【表 8】の結果が得られた。

DESPATCH から受信されたテレメトリデータのその後の解析により, DESPATCH の内部温度が想定されたよりも高くなり, ソフトウェアによる自動送信停止モードが作動していたことが確認された。内部温度が大幅に高くなることは想定外の挙動であったが, ソフトウェアによるフェールセーフは設計通りに作動したことになる。

電波の送信動作が間欠的になったことにより, 電力消費が軽減され, 当初の予定よりも大幅に長く運用を行うことができた【図 14】。最終的には, オランダ^{*40}とチェコ^{*41}のアマチュア無線家が, 469.67 万 km からの第 3 フェーズの信号の受信に成功した⁹⁾¹⁰⁾。

表 8 第 1 回目のパスの運用結果

時刻 JST	イベント
18:52	AOS, モールス符号視認
23:38	信号途絶
00:28	信号復活
00:42	ボーコード確認
00:54	信号途絶
01:36	信号復活
01:58	信号途絶
02:45	信号復活
03:00	信号途絶
03:44	LOS



図 13 多摩美地上局で受信されたボーコード

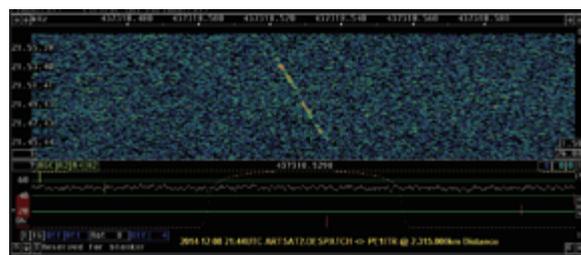


図 14 オランダで受信された 231.5 万 km からの信号

6.3 JSONRPC を利用した連携

6.3.1 artsatd と連携する IoT

artsatd が提供する JSONRPC インターフェースを利用して, ネスレ日本株式会社^{*42}が販売するバリスタ TAMA^{*43}と artsatd を連携させる実験を行った。バリスタ TAMA のメインコントローラを改造して, mbed^{*44}と WIFI モジュール^{*45}を追加し, mbed から JSONRPC を使用して artsatd に接続するネットワークバリスタ^{*46}を制作した【図 15】。

^{*40} PEIITR - <http://www.itr-datanet.com/~peiitr/artsat2despatch>

^{*41} OK1DFC - <http://www.ok1dfc.com/eme/despatch/despatch.htm>

^{*42} <http://www.nestle.co.jp>

^{*43} <http://nestle.jp/brand/nba/tama/index.html>

^{*44} LPC1768 - <http://developer.mbed.org/platforms/mbed-LPC1768>

^{*45} GS1011MIC - <http://www.sugakoubou.com/doku/gs-wifi>

^{*46} http://github.com/toolbits/tama_hack



図 15 artsatd と連携する IoT

ネットワークバリスタは、衛星や宇宙機の運用時刻を artsatd から取得し、運用が終わる時刻に自動的にコーヒーを淹れる。運用時刻を artsatd から取得するため、artsatd で現在追尾している衛星の到来に合わせた動作を行うようになっている。

6.3.2 外部のウェブサイトとの連携

ARTSAT プロジェクトのウェブサイトでは、衛星の打ち上げからの経過時間や、地上局と衛星の距離などをリアルタイムで表示している。この機能を実現するためにウェブサイトと artsatd を連携させ、ウェブサイト上の Java スクリプトから artsatd にアクセスする実験を行った。【コード 9】にウェブサイト上の Java スクリプトの例を示す。

6.4 改良点の検討

artsatd は、超小型衛星の運用を目的に開発されたソフトウェアであり、現在のところビーコン用の CW モードとデータ送受信の FM モードを通信形式としてサポートしている。衛星の種類によっては、CW をデータ通信として利用したり、音声信号を FM で送信したりと多種多様である。CW と FM 以外にも SSB をサポートする衛星も存在し、これらの通信形式にも対応できるようにデータベースを再設計し、ユーザーインターフェースの拡張を行うと良い。

現在の実装では、追尾したい衛星を 1 つ選択

して、軌道計算を行い監視する。この機能を拡張して、複数の衛星の軌道を同時に計算し、可視範囲に入る衛星が存在すればそれを追尾するように改良すると便利である。多数の衛星の軌道を常に計算するには大きな処理能力が要求されるため、適切な仕様の設計が大切である。

セキュリティ面では、localhost からのアクセスのみを許可しているが、これを拡張して firewall 内のコンピュータからもアクセスできるようにすると、ネットワークの設計が柔軟になる。この改良を行うためには、自機の IP アドレスを取得して、設定されているサブネットマスクを適用し、アクセス元の IP アドレスと比較する。

上記の改良点の他にも、衛星一覧や軌道要素のデータベースの編集機能の追加、開発された環境以外へのソースコードの移植などが今後の改良点として考えられる。

コード 9 Java スクリプトからのアクセス (一部抜粋)

```
<script type="text/javascript">
$(function() {
    countUp();
});
function countUp() {
    setTimeout('countUp()', 1000);
    $.post('http://artsat.idd.tamabi.ac.jp:16782/rpc.json',
        '{
            "jsonrpc": "2.0",
            "method": "observer.
                getSpacecraftDistance",
            "params": null,
            "id": 1
        }',
        function(data) {
            var km = data.result.distance;
            $("#Distance").html(km + '<span
                class="day"> km</span>');
        },
        'json');
    }
</script>
```

7 謝辞

ARTSAT プロジェクトは、2014 年度多摩美術大学共同研究費「超小型芸術衛星 INVADER の打ち上げと ARTSAT プロジェクトの展開」および 2014 年度科研費基盤研究 (C)「衛星芸術用ミッションモジュールの開発と遠隔創造の実践」

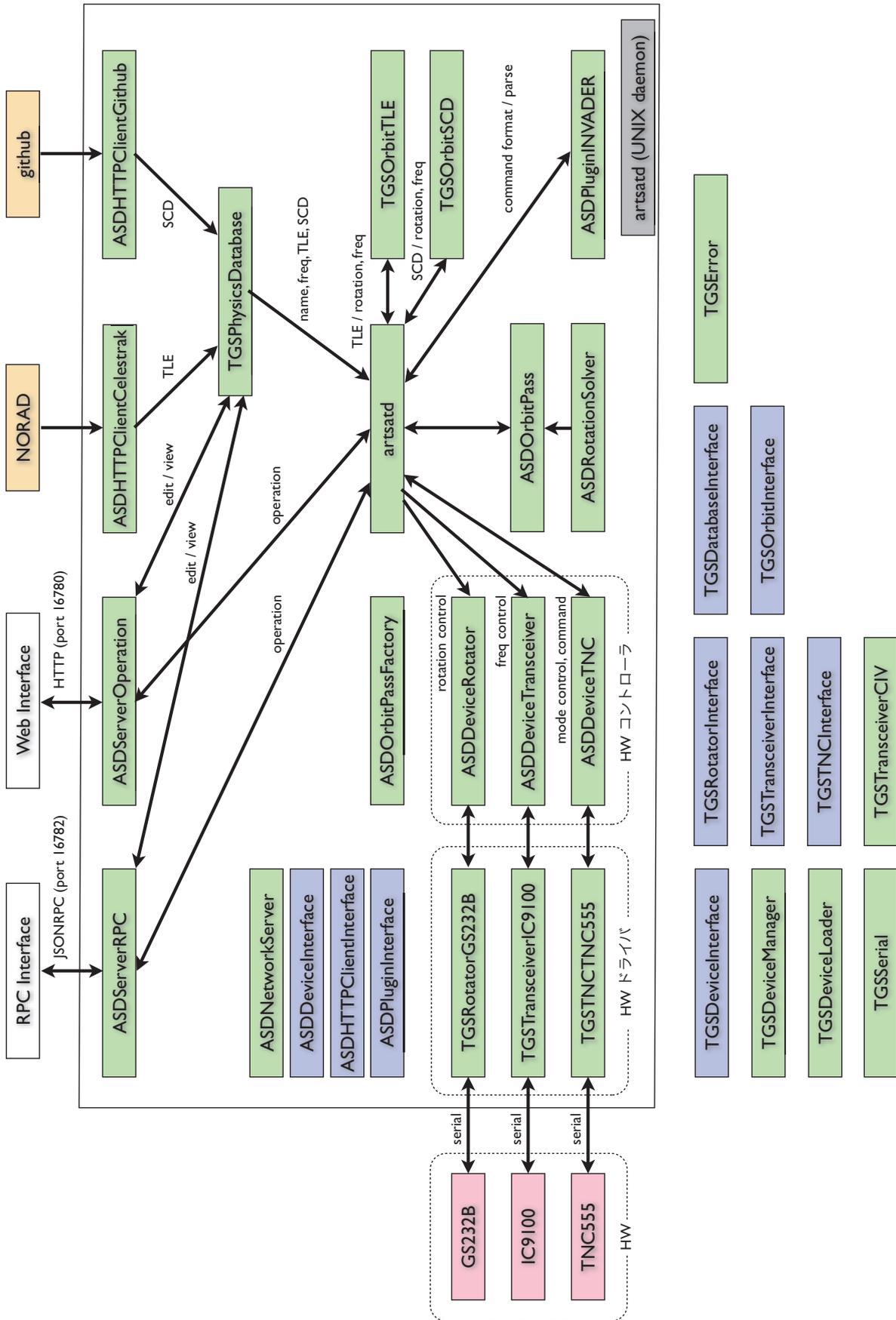
の支援を受けて進められた。

田中利樹 INVADER 開発 PM, 宇佐美尚人 DESPATCH 開発 PM, MORIKAWA の開発および多摩美地上局の設置と運用に尽力いただいた中澤賢人氏を始めとする ARTSAT プロジェクトメンバーと関係各位の尽力に厚く御礼を申し上げます。

参考文献

- 1) 宮崎康行 (著), 人工衛星をつくる - 設計から打ち上げまで -, オーム社, 2011, p.188
- 2) JSON-RPC Working Group, JSON-RPC 2.0 Specification, <http://www.jsonrpc.org/specification> (2013-01-04)
- 3) W. Richard Stevens, Stephen A. Rago (著) 大木敦雄 (訳) (監), 詳解 UNIX プログラミング, 翔泳社, 2014, 第 3 版, pp.372-376
- 4) 宮崎康行 (著), 人工衛星をつくる - 設計から打ち上げまで -, オーム社, 2011, pp.101-103
- 5) Michael D. Bauer (著) 豊福剛 (訳), Linux サーバセキュリティ, オライリー・ジャパン, 2003, pp.127-130
- 6) W. Richard Stevens, Stephen A. Rago (著) 大木敦雄 (訳) (監), 詳解 UNIX プログラミング, 翔泳社, 2014, 第 3 版, pp.435-438
- 7) Antony Polukhin, Boost C++ Application Development Cookbook, Packt Publishing, 2013, pp.138-141
- 8) 宇宙航空研究開発機構 (編), 宇宙航空研究開発機構研究開発報告宇宙科学情報解析論文誌 第四号, 弘久社, 2015, JAXA-RR-14-009, pp.26-28
- 9) Rob Hardenberg, ARTSAT2:DESPATCH Deep Space Probe Reception Report, <http://www.itr-datanet.com/~pelitr/artsat2despatch> (2014-12-15)
- 10) Zdenek Jaroslav SAMEK, RX test DESPATCH 437,318 MHz, <http://www.ok1dfc.com/eme/despatch/despatch.htm> (2014-12-14)

付録 A



第二世代EDISON (衛星運用工学データベースシステム) の開発

高木 亮治*¹, 北條勝己*¹ (宇宙航空研究開発機構)

Development of Second Generation of Engineering Database for ISAS Spacecraft Operation Needs: EDISON

by

Ryoji Takaki*¹ and Katsumi Hojo*¹

ABSTRACT

EDISON (Engineering Database for ISAS Spacecraft Operation Needs) is a database system which aims to provide engineering data related to spacecraft operations. It has HK telemetry data from spacecraft and operation status of ground systems, such as tracking data, orbital elements, command histories and so on. EDISON has following features; the available data are in physical quantities for user's convenience, the system does not use any commercial database software to reduce maintenance load for the life of the satellite, and the data in EDISON can be accessed using a web browser. A user can sample a part of the data, and download it as a text file of CSV format. The seven EDISON systems for NOZOMI, HAYABUSA, SUZAKU, AKARI, HINODE, AKATSUKI and IKAROS were constructed and are in service currently. Second generation EDISON for HISAKI and HAYABUSA2 has been developed with new concepts and technologies. This paper presents the overview of second generation EDISON.

Keywords : 工学データベース, 衛星運用, RDB

概 要

科学衛星運用工学データベースEDISON(Engineering Database for ISAS Spacecraft Operation Needs)は科学衛星や探査機の運用に必要とされる工学情報(テレメトリデータ, 局設備からのデータなど)を一元的に収集管理し, また利用しやすい形で関係者に配布するためのシステムである。EDISONでは物理量に変換されたデータを, Webブラウザを通じてCSVファイル形式で提供することでユーザーの利便性を高めている。EDISONはこれまでISASの主要な7つの衛星向けにシステムが開発され運用されている。この度新しく打ち上げられた「ひさき」, 「はやぶさ2」向けに第二世代EDISONシステムを開発したので, その概要について紹介する。

* 平成27年12月17日受付 (Received December 17, 2015)

*¹ 宇宙航空研究開発機構宇宙科学研究所 (JAXA/ISAS)

1. はじめに

宇宙航空研究開発機構宇宙科学研究所では、科学衛星の運用に関わる工学データの収集、蓄積、配信システムとして EDISON(Engineering Database for ISAS Spacecraft Operation Needs)¹⁾ と呼ばれる衛星運用工学データベースシステムを開発運用している。これは、人工衛星や探査機の運用に必要とされる工学情報（テレメトリデータ、局設備からのデータなど）を一元的に収集管理し、また利用しやすい形で関係者に配布することを目的とし、衛星・探査機の運用及び異常時の原因究明の支援、観測データ解析のための基本的なデータ供給といった事が期待されている。EDISON の特徴としては、衛星・探査機の運用や工学データ解析に有用な各種データの一括管理、提供する全てのデータは物理量に変換した形で提供、データは Web インターフェイスで提供といった点が挙げられる。取り扱うデータとしては HK テレメトリデータを中心に、地上受信局の設備監視データ、追跡データ、軌道要素、アンテナ予報値、アンテナ可視、コマンド履歴など多岐に渡る。提供されるデータは全て工学値と呼ばれる具体的な物理量に変換した形で提供されるため、ユーザーは工学値変換を行うことなく直ぐにそのデータを利用することが可能である。これらのデータは Web インターフェイスを通じて取得することができるため、インターネットに接続した端末であれば、特別なクライアントソフトをインストールする必要なく、一般的な Web ブラウザを用いてデータの検索・取得が可能となっている。また、これらの有益なシステムが継続的に衛星運用を支援することが可能となるために開発・維持管理コストもできるだけ抑ええた設計となっている。

EDISON はこれまでに、火星探査機「のぞみ」、小惑星探査機「はやぶさ」、X 線天文衛星「すざく」、赤外線天文衛星「あかり」、太陽観測衛星「ひので」、金星探査機「あかつき」、小型ソーラー電力セイル実証機「イカロス」向けの EDISON が開発され運用を行っている（現在「のぞみ」用 EDISON は運用を終了し、データはオフラインで管理を行っている）。これらの EDISON は第一世代の EDISON として開発・運用されてきた。近年打ち上げられた惑星分光観測衛星「ひさき」、小

惑星探査機「はやぶさ 2」向け EDISON の開発を契機に、システムの見直しを行い、新しいやり方で第二世代 EDISON の開発を行った。ここでは、第一世代 EDISON の概要を紹介した後、第一世代 EDISON の課題を解決するために新しい方式を採用した第二世代 EDISON の概要について紹介する。

2. 第一世代 EDISON の概要

2.1 システムの概要

第一世代 EDISON のシステム構成を図 1 に示す。第一世代 EDISON では EDISON サブネット上に Proxy サーバーを入口として配置し、そのバックエンドにそれぞれの衛星毎の EDISON (物理サーバー) が独立に移働している。衛星毎の EDISON の例として「はやぶさ」用 EDISON のシステム構成を図 2 に示す。

テレメトリデータは生データを SIRIUS^{2, 3)} と呼ばれる 0 次データアーカイブシステムから取得され、テレコマデータベース (SIB: Spacecraft Information Base) に基づいて必要なテレメトリ項目データの抽出、工学値変換を実施し EDISON 内に蓄積される。一方、設備監視データなどテレメトリデータ以外のデータ（工学値変換の際に必要なとなる SIB も含む）はデータ蓄積と呼ばれる短期蓄積・配信システムから取得され、必要に応じて工学値変換を行い EDISON 内に蓄積される。例えば、設備監視データはデータ蓄積から生データとして提供されるため EDISON 内で工学値変換され蓄積されるが、軌道データ、コマンド履歴などのデータは工学値変換済みデータとして提供されるため EDISON 内にそのまま蓄積される。個々の衛星毎に細かな部分は異なるが、大枠としては同じシステム構成となっている。

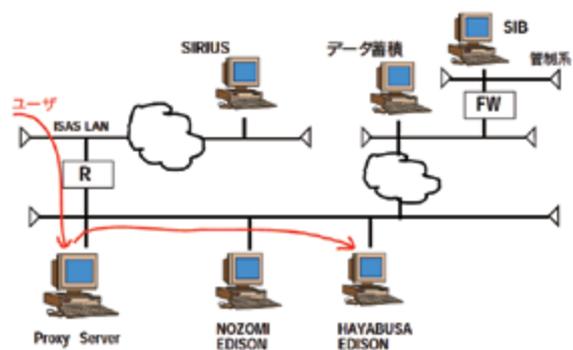


図 1. 第一世代 EDISON システムの構成図(「のぞみ」用 EDISON, 「はやぶさ」用 EDISON を例示)

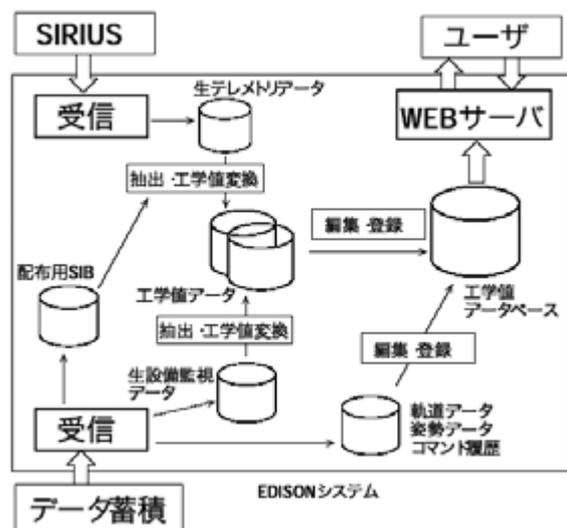


図2. 「はやぶさ」用 EDISON のシステム構成図

2.2 特徴と課題

第一世代 EDISON の特徴としては以下の点が挙げられる。テレメトリデータに関しては全てのデータを EDISON で蓄積するのではなく、利用頻度の高いデータを厳選して蓄積している。テレメトリデータ以外のデータである地上局データ、姿勢、軌道データなども各衛星の目的、運用ポリシーなどに合わせて選択して蓄積した。また、高度な検索機能は付加せずに簡素なシステムを目指した。またデータのセキュリティに関しても最低限のアクセス制御としてグループアカウントでの管理を採用した。

これらの特徴は EDISON の開発を開始した当時の状況に起因するものである。例えば、蓄積するデータを厳選したのは当時のハードウェアの制限によるものであり、全データを蓄積するためには高価な大容量ハードディスクが必要であったため、それを避けるためにデータを厳選することで対応した。その結果、蓄積すべきデータを厳選するための衛星プロジェクトとの調整が必要になると同時に、EDISON を構成するソフトウェア上でもそのリストをコーディング管理する必要が発生するなどソフトウェアの管理コストが増大する要因となった。そのためシステム的には衛星毎の個別対応がかなりのウェイトを占めることとなり、実質的には衛星毎に一品物に近いシステム開発となった。また、データのセキュリティに関しても開発当時は昨今とは比べ物にならないくらい大らかな対応で十分であったが、昨今は非常に厳しくなり、データセ

キュリティへの確実な対応が大きな負担となってきた。

3. 第二世代 EDISON の開発

3.1 開発の狙い

前節で紹介したように、第一世代 EDISON では開発を開始した当時の制約や状況の下で設計・開発を行ってきており、現状ではいくつかの課題を抱えている。そのため、新しく打ち上げられる衛星向けの EDISON の開発を契機にシステムの見直しを行い、第二世代 EDISON の開発を行った。その際に EDISON 本来の目的に立ち戻り、機能要求の整理を行った。現時点で EDISON に要求される機能としては、データを取得し工学値に変換する、データを蓄積する、主に WebI/F を経由してデータを配信する、となる。これらの機能要求をベースに第一世代 EDISON 課題であった

- ① 衛星個別対応リストの削減
- ② セキュリティ強化
- ③ 機能強化

の解決を第二世代 EDISON での開発のねらいとした。①に関しては共通化ツールの導入、システム構成の整理による更なる開発・運用コストの削減、②に関しては最新 Web 技術の導入によるセキュリティ強化、③に関しては機能強化への布石として RDB(Relational DataBase) の導入を実施した。

まず、全体的な話として、システム構成の整理を行った。これはユーザーサービスの視点からシステムを整理したもので、システム構成をミニマム構成とオプション構成(機能)に整理した。ここでは最低限必要な機能をミニマム構成と定義し、システムの再開発を実施した。またオプション機能はより高度な機能として開発し、実用的な評価を経てミニマム構成へ取り込むという方針で機能強化を図ることとした。図3に第二世代 EDISON のシステム構成を示す。第二世代 EDISON ではシステム構成が「ミニマム構成(機能)」、「オプション構成(機能)」に整理されている。取り扱うデータは、テレメトリデータを工学値に変換した結果である「CSV ファイル群」とテレメトリデータ以外のデータファイルの「その他ファイル」であり、ユーザーインター

フェイスは「Web I/F」、「SQL I/F」がある。「Web I/F」には登録されたファイルの書誌情報の検索である「ファイル検索機能」とRDBに登録された工学値の検索機能である「検索機能」がある。これらの詳細に関して機能毎に順次説明する。

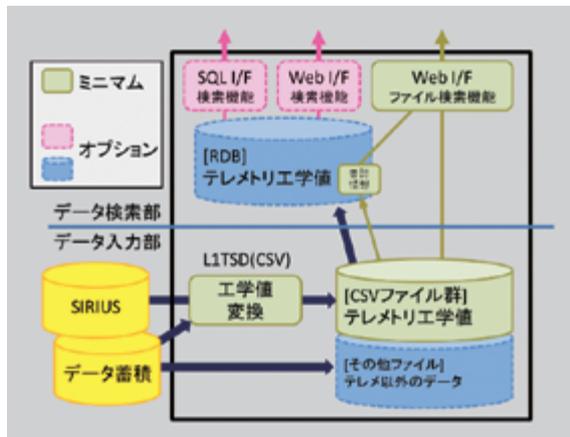


図3. 第二世代 EDISON のシステム構成図

3.2 データ取得・変換

テレメトリデータは SIRIUS から生テレメトリデータを取得し工学値に変換する。第一世代では衛星固有の変換プログラムを利用していたが、この部分を L1TSD（レベル1時系列データフォーマット変換ツール）⁴⁾CSV版と呼ばれる共通化ツールを使うこととした。L1TSDを使うことで EDISON ソフトウェアにおける各衛星への個別対応の必要がなくなり、各衛星プロジェクトが用意するテレコマデータベース (SIB) をもとにテレメトリデータを工学値に変換することが可能となった。L1TSD CSV版では指定された時間のテレメトリデータを SIRIUS から取得し、APID 毎にかつ1時間もしくは1日単位で CSV ファイルに出力する。また、従来はテレメトリ項目を厳選して変換・蓄積したが昨今の H/W の価格破壊により十分な量のディスクが確保できるようになったため、全てのテレメトリデータ (HK 系のデータではあるが) を対象とし変換・蓄積することとした。テレメトリデータ以外のデータに関してはデータ蓄積から簡単なスクリプトで必要なファイルを取得することとした（「その他ファイル」）。

このようにデータの取得・変換に関しては共通化ツール L1TSD の導入および全テレメトリデータの登録により衛星毎の個別対応が不要となり、個別対応に起因するソフトウェアの開発・

システムの維持・管理作業を削減することができた。その結果コストを抑えつつ全 HK テレメトリデータを対象とするなどサービスの向上が可能となった。

3.3 データ蓄積

工学値に変換されたテレメトリデータは CSV ファイルとして蓄積される。それ以外の「その他ファイル」もファイル単位で蓄積される。これらのファイル情報は書誌情報として PostgreSQL（以後 RDB と呼ぶ）を用いて管理している。RDB への書誌情報の登録も、読み込みディレクトリの指定、検索キーとなる時刻をファイルの命名規則で陽に指定することで、衛星毎の差異に煩わされることなく共通的に処理する事を可能とした。また、オプション機能ではあるが工学値を RDB に登録する機能もある。テレコマデータベースから RDB のテーブルを自動生成するツールを開発した。これらの機能により、衛星個別対応を排除し、全自動でデータ登録・蓄積が可能となった。

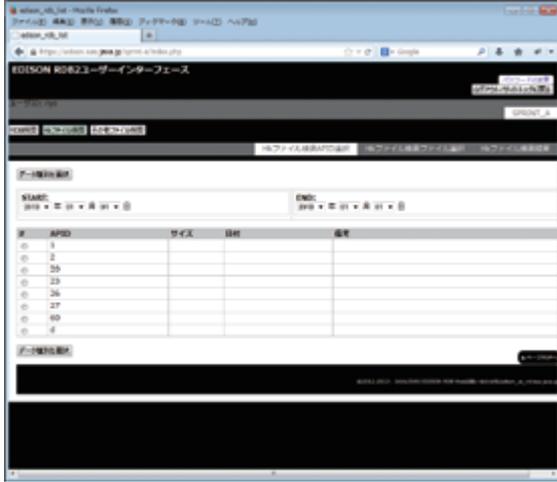
3.4 I/F を経由したデータ配信

Web I/F を使ってデータを配信する部分は PHP を用いて最低限の作りこみを行った。その際にセキュリティ強化のためにユーザーの個別アカウント、データアクセス制御、ユーザーによるパスワード管理機能などを実現した。

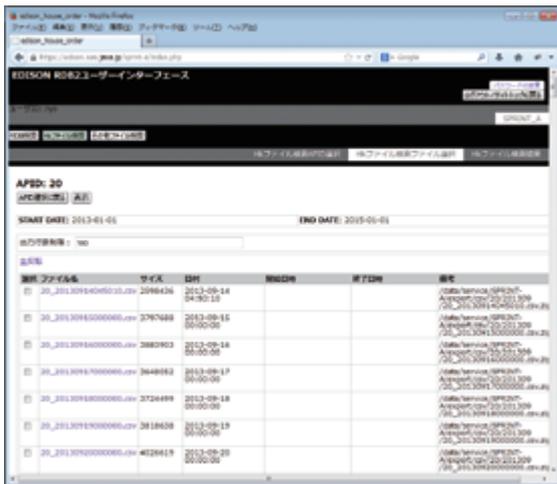
ミニマム構成では Web I/F 経由でファイル書誌情報の検索（期間、APID を指定した検索）が可能となっている。例として図4に CSV ファイル検索画面（図4.a APID および期間の指定、図4.b ファイル指定）を示す。

「その他ファイル」に関してはファイルの命名規則を設定することで、自動的に検索画面が作成できる。図5に自動生成された「その他ファイル」の検索画面を示す。また、オプション機能として RDB に登録された工学値への I/F として Web I/F および SQL I/F を用意した。Web I/F ではテレメトリ項目名、期間、サンプリング間隔を指定した検索が可能である。これらの検索条件はファイルとしてユーザー毎に保存 / 読込することが可能となっている。SQL I/F では直接 SQL を使うことで高度な検索が実施できる。SQL I/F を活用することで、Web I/F 経由でも

高度な検索機能を比較的簡単に実現できると考えている。これらの機能によりセキュリティの強化、さらに衛星毎の作りこみの排除を実現した。



a) APID および期間指定の画面



b) ファイル指定画面

図 4. CSV ファイル検索画面

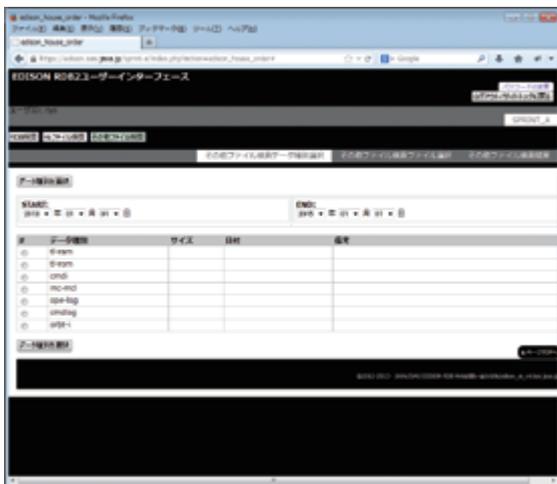


図 5. 自動生成された「その他ファイル」の検索画面

3.5 ユーザーサービスとしての視点

第二世代 EDISON をユーザーサービスの視点で見ると Web I/F で以下のサービスを実現したことになる。

- ① CSV ファイル形式の HK 工学値データを取得できる。その際に APID, 期間を指定した絞り込みが可能。
- ② その他ファイル (テキストに限らない) を取得できる。その際にファイル種類, 期間を指定した絞り込みが可能。
- ③ HK 工学値データを取得できる。その際に APID, テレメトリ項目名, 期間, サンプリング間隔を指定した絞り込みが可能。また, SQL I/F を使った高度なデータ検索が可能。但し, ③に関してはオプション機能としてパイロット的に実施したが, 現時点では残念ながら性能上の問題, つまり速度が遅すぎて実用に耐えられない状況である。いくつかの改善を試みているが実用的な性能を実現できていない。抜本的な取り組みが必要と思われる。

4. おわりに

第一世代 EDISON の開発・運用から得られた知見を基に第二世代 EDISON の開発を行った。

- ・ L1TSD 等の共通化ツールの導入, 全 HK テレメトリの蓄積対象化, ファイルの命名規則を前提とした自動化ツールの開発により衛星毎の個別対応を極力排除し, サービスを必要最低限に整理することで新衛星向け EDISON の開発コストを大幅に削減することができた。
- ・ 各種自動化ツールを開発することでほぼ全自動での運用が可能となり運用コストも削減することができた。
- ・ 最新 Web 技術の導入により個別アカウントによるセキュリティ強化を実現した。
- ・ 検索機能高度化の土台として SQL を導入した。今後より便利なサービスを低コストで実現可能と考えている。

今後の課題としては RDB のデータ登録および検索性能であり, 現時点では実用に耐えられない性能 (衛星 1 日分のデータ登録に数日かかる) である。高速化に向けて並列処理の導入, 登録/検索アルゴリズムの見直し, バッチ処理の導入などを検討している。

参考文献

- (1) H. Honda, R. Takaki, A. Choki, M. Hashimoto and T. Hirose, Engineering Database for ISAS Spacecraft Operation Needs (EDISON) for Low Earth Orbit Satellites, ISTS 2006-f-26, 2006.
- (2) 岡田, 馬場, 松崎, 山本, 永田, 飯塚, 岩澤, 科学衛星データレベル0 処理システムの機能改修, 第11回宇宙科学シンポジウム, P7-087, 2011.
- (3) 岡田, 山本, 次期科学衛星テレメトリデータベースに関する検討, 宇宙科学情報解析論文誌第1号 (JAXA-RR-11-007), pp151-156, 2013.
- (4) 松崎, 高木, 山本, 篠原, レベル1時系列データフォーマット変換ツール (FITS) の開発, 平成26年度宇宙科学情報解析シンポジウム, 2015.

レベル1時系列データフォーマット変換ツールの開発

松崎恵一^{*1*2} 山本幸生^{*1*2} 高木亮治^{*1*2} 篠原 育^{*1*2}

Development of Level-1 Time Series Data Format Conversion Tool

Keiichi MATSUZAKI^{*1*2}, Yukio YAMAMOTO^{*1*2}, Ryoji TAKAKI^{*1*2}, Iku SHINOHARA^{*1*2}

Abstract

Processing of level 1 time series data was realized by software programs which take into account individual file specifications. In previous space science projects, these programs were manufactured with certain amount of effort in every spacecraft projects. In order to minimize such efforts, we developed a general program - Level-1 Time Series Data Format Conversion Tool (L1TSD FCT) which is configured by Spacecraft Information Base (SIB) which describes specification of telemetry data and template file which describes specification of output file. This program performs memory base data processing in high speed and is used in data processing in Japanese space science projects since HISAKI.

概要

レベル1時系列データの処理を実現するには、従来、作成するファイルの書式に合わせ処理プログラムを開発する必要があり、労力を要していた。我々は、これを、より少ない労力で実現するため、衛星のテレメトリ設計を記述したSIB (Spacecraft Information Base; SIB) と出力の書式を規定するテンプレートを読み込み1次データ処理を実現する汎用な処理プログラム、レベル1時系列データフォーマット変換ツールを開発した。このプログラムは、オンメモリでデータ処理を行うことで高速に動作する。このプログラムは、ひさき以降の日本の科学衛星のデータ処理に適用されている。

Keywords: データ処理, L1TSD, SIB

1. はじめに

宇宙科学を進めるには、衛星からのテレメトリを科学者が使えるような形に加工するデータ処理が必要である。このデータ処理は、極力少ない労力で実現されるのが望ましい。そのため、我々は、衛星プロジェクト間で処理プログラムの共通化に取り組んでいる。

プロジェクトで実施しているデータ処理を比較検討したところ、最も共通性が高いものとして、HK (House Keeping) や機器ステータスなど固定長の時系列のデータを物理量に変換する処理が洗い出された。物理量に変換されたテレメトリは、しばし、レベル1データと呼ばれる。以下では、時系列からなるレベル1データを、特に、レベル1時系列データ (Level-1 Time Series Data; L1TSD) と呼ぶ。

レベル1時系列データは、衛星運用やデータ解析などで使用される。図1に、従来のISAS (Institute for Space and Aeronautical Science) の衛星運用・データ処理システムにおける、レベル1時系列デー

* 平成27年12月17日受付 (Received December 17, 2015)

^{*1} 宇宙航空研究開発機構 宇宙科学研究所
(Institute for Space and Aeronautical Science, Japan Aerospace Exploration Agency)

^{*2} 総研大 物理科学研究科 宇宙科学専攻
(SOKENDAI, The Graduate University for Advanced Studies, School of Physical Science, Space and Aeronautical Science)

タの作成（一次データ処理）の登場箇所を示す。一次データ処理は、運用システムを構成する汎用衛星運用試験ソフトウェア（Generic Spacecraft Test and Operation Software; GSTOS）¹⁾ や、工学データベース²⁾・科学データベース³⁾ 向けの処理に登場する。これらの処理結果は同一であることが期待されるが、従来、別々の処理プログラムを用いて実現されてきた。このうち、運用システム・工学データベースでは、各々の範囲内で、衛星プロジェクトを跨いだ処理プログラムの共通化が図られていたが、科学データベース向けの処理プログラムは、衛星プロジェクトが個別に開発しており、毎回、プログラム作成のコストがかかっていた。

この非効率な状況を改善するため、我々は、工学データベース・科学データベース共通で1次データ処理を実現するプログラム（レベル1時系列データフォーマット変換ツール）を作成することとした。レベル1時系列データフォーマット変換ツールは、工学データベースや各種の学問分野に対応するため、幾つかの出力フォーマットに対応することとした。本論文は、このレベル1時系列データフォーマット変換ツールの開発結果について示す。

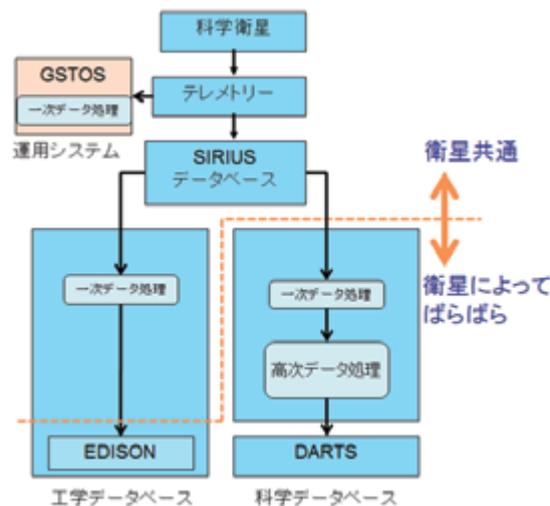


図1 ISASにおける従来のレベル1時系列データ処理

2. 従来の科学データベース向け1次データ処理プログラム

2.1 衛星情報ベース (SIB)

ISASの衛星プロジェクトでは、テレメトリ・コマンドの設計情報を、衛星情報ベース (Spacecraft Information Base; SIB) と呼ばれるデータベースに記述し、GSTOSなど、汎用な衛星の運用・試験ソフトウェアに読み込ませ動作させることで、衛星毎の運用・試験ソフトウェアの開発費の低減を図っている。搭載機器のテレメトリ設計は、メッセージ内のテレメトリ項目がメッセージ毎に変わる場合 (サブコミなどとも呼ばれる) があるなど多様であり、SIBもこれに対応したものになっている。表1に、SIBで定義可能なテレメトリ設計の自由度の概要を示す。SIBは当初はHKデータのみをスコープとし、固定長のテレメトリにのみ対応している。近年はミッションデータにも使用されるが、可変長のデータは依然、記述の対象外である。

表 1 SIB で定義可能なテレメトリ設計の自由度の概要

抽出定義	無条件抽出：固定のあるワード位置から無条件にビットパターンを抽出 条件付き抽出定義：他のテレメトリ値が一定の条件を満たす際（サブコミ - 時間分解能を間引き，複数の項目で一つのワード位置を使用 - など）にビットパターンを抽出 ダミー抽出：ビットパターンは抽出しないがテレメトリ項目の処理を開始する（後段の変換処理で他のテレメトリ項目を参照し値を定める）
エンコーディング	符号無し整数，符号有り整数，単精度浮動小数点，倍精度浮動小数点
工学値変換	無条件変換：必ず適用 条件付き変換：他のテレメトリ値が一定の条件を満たす際に適用
変換方法	1) 多項式変換：5 次までの多項式の係数を指定する方法 2) 任意の関数式による変換： $f(x) = \log(x)/x$ など任意の関数式を指定する方法 3) プログラムを用いた変換：C 言語のプログラムを用いる方法 2) 3) では，他のテレメトリ値を参照し，値の合成が可能

※下線は，従来の科学データベース向けレベル1時系列データの処理プログラムがサポートしていた範囲。

2.2 標準フォーマット

近年，科学コミュニティに向けたデータ配布では，研究者が容易に扱うことができる標準フォーマットによる公開・配布が一般的となってきている。これらの標準フォーマットは，ファイルを見ただけで，ある程度で内容を理解できるよう，自己記述的 (self-descriptive) である。例えば，天文学やその周辺分野では FITS (Flexible Image Transport System) フォーマット⁴⁾ が用いられ，太陽地球系科学では CDF (Common Data Format)⁵⁾，惑星科学分野では PDS フォーマットや SPICE フォーマット⁶⁾ が用いられる。

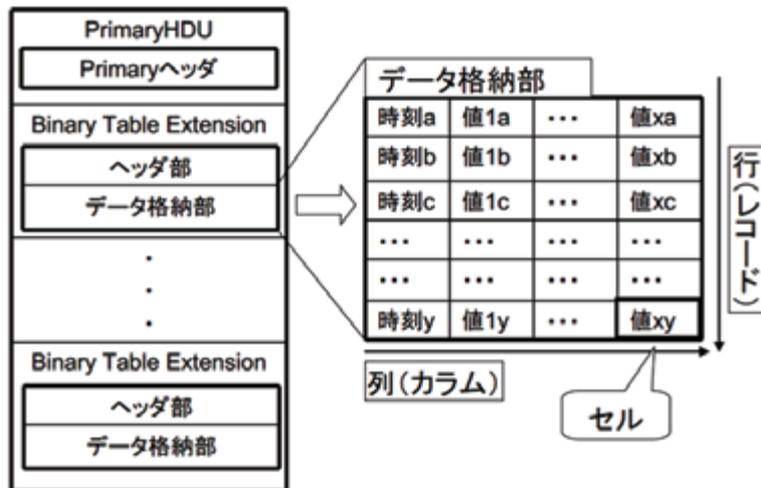


図 2 FITS ファイルへの時系列データ格納イメージ

[「レベル1時系列データフォーマット変換ツール(FITS)ユーザズガイド」より引用]

例として図 2 に，FITS ファイルへの時系列データ格納イメージを示す。FITS ファイルは，Primary HDU (Header Data Unit) の後に 複数の Extension を持つ。いずれの Extension も，データ格納部に如何にデータを格納するかはヘッダ部にて規定される。ファイル全体はバイナリ形式だが，ヘッダ部は固定長 (80byte) の ASCII 文字の繰り返しとなっている。Extension には，Binary Table

Extension (BTE) のほか、ASCII Table Extension, Image Extension などバリエーションがあるが、時系列データをよりコンパクトに格納するには、カラムごとにバイナリ形式のデータ型を指定できる BTE を使用する。

図3に FITS ファイルを用いる場合の書式の規定の例を示す。FITS の書式は、この例のように、ヘッダの例を用いて記述されることが多い。BTE では、カラム毎に、TTYPEn, TUNITn, TFORMn などのキーワードを用い、それぞれ、カラムのラベル、単位、フィールドの型とサイズなどを規定する。TFORMn の値に登場する、1B, 1I, 1J, 1D などの値は、それぞれ、スカラー型の符号無し 8bit 整数、符号あり 16bit 整数、符号あり 32bit 整数、倍精度浮動小数点数に対応する。ヘッダには、その他、観測情報など任意のキーワードに対し、値とコメントを記述できる。

2.3 従来のレベル1時系列データの処理確立のステップ

FITS などこれらのフォーマットに準拠し、データをどのような書式で配布するかは、衛星プロジェクト毎に設計する。書式を規定した後、実際にテレメトリデータを格納したファイルを作成するには、それぞれに対応した処理プログラムを作成する。表2に、従来のレベル1時系列データの処理確立のステップを示す。(a)に示す古典的な手法は、前提知識の獲得のステップ以外は、扱うデータ種別の量に比例した作業が発生する。典型的なテレメトリの項目数は、従来、衛星全体で、数千のオーダーでありプログラム作成と検証の作業負荷が高かった。これに対し、(b)に示す SIB を用いる手法は、書式に規則を設け、その規則に対応した汎用プログラムを書けばデータ種別に比例した作業量を抑えられるというメリットがある。検証の規模は、テレメトリ項目数の規模から、データの種別数(数十)と減るが、他方で SIB を扱った汎用プログラムを組む必要があるため、プログラム作成の難易度がより高くなる。また、開発コストの元を取るには、衛星プロジェクト全体で一つのプログラムを使うなど、まとまった範囲への適用が必要となっていた。そこで、HK など SIB2 で記載できるデータについても、(a)の手法で、個別の衛星プロジェクトでのプログラミングを行うことがあった。(b)の手法は、原理的に SIB のあらゆる定義をサポートすることが可能である。しかし従来の ISAS プロジェクトで開発されたレベル1時系列データの処理プログラムの多くは、SIB がサポートするテレメトリ設計のうち、メジャーなもの(表1の下線部)しかサポートできていなかった。搭載機器設計がこれを超えており、処理が必要な場合には「古典的な方法」でプログラムを作成することで対応されてきた。

```

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789
-----
SIMPLE = T / DATA IS IN FITS FORMAT
BITPIX = 32 / 32 BITS TWOS COMPLEMENT INTEGERS
NAXIS = 2 / NUMBER OF AXIS
NAXIS1 = 1024 / PIXELS ON 1st MOST VARYING AXIS
NAXIS2 = 1024 / PIXELS ON 2nd MOST VARYING AXIS
EXTEND = F / Existence of extension or not
BLANK = 32768 / Value used for NULL pixels
BSCALE = 1.00 / Real = fits-value*BSCALE+BZERO
BUNIT = 'ADU' / Unit of original pixel values
BZERO = 0.00 / Real = fits-value*BSCALE+BZERO
COMMENT
COMMENT ***** Telescope, Instrument and FITS header version
COMMENT
OBSERVAT= 'NAOJ' / Observatory name
TELESCOP= 'SUBARU' / The name of telescope data obtained
INSTRUME= 'CIAO' / The name of instrument
OBS-ALOC= 'Observation' / Allocation mode for Instrument
INS-VER = CIAO-H01S02 / version of the instrument / control-soft
COMMENT
COMMENT ***** Observation *****
COMMENT
OBSERVER= 'CIAO' / Observer
PROP-ID = 'o11227' / Proposal ID
DATE-OBS= '2011-10-17' / Observation start date ('yyyy-mm-dd')
UT = '03:58:55.642' / HH:MM:SS.S start UTC at exposure
HST = '17:58:55.642' / HH:MM:SS.S start HST at exposure
LST = '20:40:48.382' / HH:MM:SS.S start LST at exposure
MJD = 52950.16591746 / Modified Julian day
TIMESYS = 'UTC' / Time system used in this header
DATASET = 'DS0000' / ID of an observation dataset
FRAMENB = 'CIAA00001707' / Image sequential number
    
```

図 3 FITS ファイルの書式の規定の例
[参考文献 7) より引用]

表 2 従来のレベル 1 時系列データの処理確立のステップ

(a) 古典的な手法	(b) SIB を用いる手法
<ul style="list-style-type: none"> - 搭載機器の設計書を読み解きテレメトリのフォーマットを理解する - 工学値への変換式を理解する 	<ul style="list-style-type: none"> - (SIB の書式を理解する) - SIB を読み込み動作する汎用なプログラムを作成する
<ul style="list-style-type: none"> - (FITS の規格を理解する) - FITS に従い、ファイルの書式を規定する - (C 言語など、一般的なプログラムの書き方を覚える) - (FITSIO など、FITS を扱うためのライブラリの利用法を覚える) 	
<ul style="list-style-type: none"> - 書式ごとにプログラムを書き下す 	
<ul style="list-style-type: none"> - 書式ごとに処理結果が正しいか検証する 	

(...) で示した部分は、前提とする知識であり、担当者が既獲得ならば省くことができる。

以上をまとめると、Level1 時系列データを処理するプログラムを作成するには、「古典的な方法」「SIB を用いる方法」が存在したが、いずれの方法もプログラミングやテレメトリ処理（ビットパターンの抽出、工学値変換など）の知識が必要とされていた。特に、「古典的な方法」は、テレメトリの規模に比例したプログラミングが必要だった。また、「SIB を用いる方法」は、汎用なプログラムのため難易度が高く、SIB の定義の全てのパターンをサポートしたものは存在しなかった。この他に、科学データベース向けにはプログラムは、衛星プロジェクト毎にプログラム開発が行われ、その全て（または、一部）が研究者により作成され、長期のメンテナンス性にリスクも存在していた。レベル1 時系列データフォーマット変換ツールは、これらの問題の解決も図るものである。

3. レベル1 時系列データフォーマット変換ツールの開発

3.1 開発のスコープの設定

2章で述べた課題を解決するため、我々は、レベル1 時系列データフォーマット変換ツールについて、従来の (b) の手法を発展させ、表3 に示す開発方針を設定することとした。

表3 レベル1 時系列データの処理プログラムへの開発方針

- | |
|---|
| <ol style="list-style-type: none"> 1) SIB で記述可能な範囲を処理対象とし、1次データ処理の担当者がプログラムを書かずとも、フルセットの SIB のテレメトリ処理を可能とすること 2) いずれのデータフォーマットに対応するプログラムも、各種のフォーマットに依存する部分を除き共通化されていること 3) 長期にわたりプログラムが維持可能なように、設計・製造・試験に必要な情報を維持・管理できる体制の下に開発を進めること |
|---|

次に、レベル1 時系列データフォーマット変換ツールが出力するフォーマットの選定を行った。候補に上げたのは、工学データベース向け：CSV (Comma Separated Values)、科学データベース向け：天文分野：FITS、太陽地球系科学：CDF、惑星分野：PDS (Planetary Data System)⁸⁾、SPICE の5つである。このうち、PDS については、それ自身フォーマットというよりは、FITS や CSV などの形式のデータに付与されるメタ情報であり、CSV や SPICE、FITS のファイルが得られれば十分であるとの分析から開発の対象から除外された。また SPICE は、幾つかのデータ種別が存在するが、衛星の時刻付けの関係を記述する SCLK (Spacecraft Clock) と、衛星の姿勢を記述する CK (C-matrix Kernel) をサポート対象に選定した。ただし、SCLK については、衛星からのテレメトリを扱うものではないので、本論文の対象からは除外する。

CDF は、FITS のような自己記述型の書式であり、今後の太陽地球系科学を考えるとサポートすることが望ましいと判定された。ただし、ISAS の今後5年を見通すと CDF を使用する衛星プロジェクトが ERG 衛星を除き登場しないことから、費用対効果を考慮し、テレメトリから CDF の直接作成はスコープ外とし、レベル1 時系列データフォーマット変換ツールの出力する CSV を CDF に加工するプログラムを ERG プロジェクトにて作成することで対応することとした。

3.2 設計概要

図4 に、レベル1 時系列データフォーマット変換ツールの構成の概要を示す。レベル1 時系列データフォーマット変換ツールは、衛星からのテレメトリ（レベル0 データ）を読み込み、レベル1 データを出力する。前段の処理では、テレメトリデータを読み込み一次データ処理の演算を実施する。この処理は SIB を読み込み動作する。SIB は、衛星開発において作成されるため、データ処理の担当者は入手し、プログラムに読み込ませるのみで良い。後段の処理では、必要に応じ補間などの処理を行うと共に、各種のフォーマットへの変換を行う。この処理は、そのフォーマットに対応したテンプレートを読み込み、補間など共通の処理を行

う。テンプレートは、データ処理の担当者にて作成する。

前段の処理、つまり、ビットパターンの抽出、工学値変換などのテレメトリ処理は GSTOS で実績がある高速な工学値変換エンジンをライブラリ化し、利用することで実現した。このライブラリは、GTAPI (GSTOS API) と呼ばれる。

後段処理、特に補間を行うには、全てのデータをメモリ上に保持するとプログラムの作りが簡単となる。このデータ保持は、独自のプログラムを組む、各種のデータフォーマット用のライブラリを使用するなどの方法があるが、比較検討した結果、我々は、CSV などを含めいずれのフォーマットに対しても、内部のデータ保持として、i) 各種のバイナリデータ型を扱え、ii) オンメモリで高速に動作し、iii) オブジェクト指向の簡便な API が用意されており、我々から開発者へのアクセスが容易な SFITSIO⁹⁾ を用いることとした。ここで、SFITSIO は、FITS のデータ構造を扱うためのライブラリである。そこで、実際には、データを読み込んだ段階で一旦 FITS が有するデータ型への変換が実施される。FITS 版では、その際に最終的な出力フォーマットに含まれるデータ型まで変換を実施する。なお、SFITSIO では、ライブラリ固有なキーワードを生成する場合があります、これらのキーワードが許容されない場合、下流のプログラムで生成されたキーワードの削除が必要となることがある。

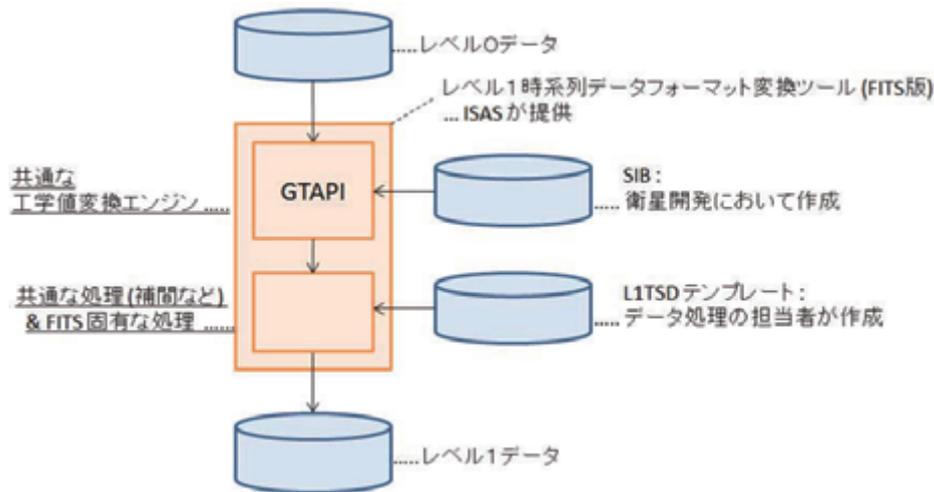


図 4 レベル 1 時系列データフォーマット変換ツールの構成の概要

3.3 レベル 1 時系列データフォーマット変換ツールを用いた処理確立のステップ

表 4 に、レベル 1 時系列データフォーマット変換ツールを用いる場合の 1 次データ処理構築のステップを示す。表 2 と比較し、必要なステップが格段に削減されていることが分かる。特に、レベル 1 時系列データフォーマット変換ツールを用いる場合、プログラムは ISAS から提供されるため、データ処理担当者はプログラミングの知識を有していなくて良いことが特徴である。データ処理の担当者には、FITS に従い、ファイルの書式を規定するという本質的な作業のみを行えば良く、3.2 節で示した GTAPI を用いることと合わせ、表 3 で示した方針の 1) を満たしていることが分かる。

表 1 レベル 1 時系列データフォーマット変換ツールを用いた場合の 1 次データ処理構築のステップ (FITS 版の例)

- | |
|---|
| <ul style="list-style-type: none"> 4) (FITS の規格を理解する) 5) FITS に従い、ファイルの書式を規定する 6) 書式ごとに処理結果が正しいか検証する |
|---|

(...) で示した部分は、前提とする知識であり、担当者が既獲得ならば省くことができる。

4. レベル1時系列データフォーマット変換ツールの機能

本章では、レベル1時系列データフォーマット変換ツールの機能について述べる。

4.1 テンプレート

レベル1時系列データフォーマット変換ツールの設計においては、プログラムが使用される可能性が極力広くなるように、それぞれのデータ処理担当者が自由に書式を規定できるようにした。例として、図5にL1TSDテンプレート(FITS版)を示す。この記法は、FITSを扱うデファクトスタンダードなライブラリCFITSIO¹⁰⁾で使用可能なテンプレートファイルを拡張した形式であり、FITSのヘッダとほぼ同じ見栄えである。そこで、FITSの知識のある人なら、誰でも記述できる。レベル1時系列データフォーマット変換ツール固有な指定として、カラムごとに、対象とするテレメトリ項目名(TTNAM#)、工学値変換するか否か(TCONV#)などを記述し、テレメトリ処理の内容を定めることとした。また、データの取得開始、終了時間などヘッダに観測情報を自動で値を設定するディレクティブ(@@ディレクティブ名)を用意した。

```

#
# SAMPLE: FILE(L1TSD)Template
#
TIME-EPH = 2005-01-01T00:00:00
XTENSION = BINTABLE
EXTNAME = BTE1
SMPLBASE = DR_A.HB_MON
TXFLDKWD = 'TTNAM,TINPL,TCONV,TSPAN,TSTAT'
DATE-CRT = @@CREATE-DATE / date of the create
DATE-OBS = @@RP-FRONT-DATE
DATE-END = @@RP-END-DATE / date of end of observation
TSTART = @@TSTART / total seconds of the DATE-OBS
TSTOP = @@TSTOP / total seconds of the DATE-END
FILENM = @@RPT:INFILENAME / file name of the RPT
HISTORY @@RPT:[1]:HISTORY
CHECKSUM =
DATASUM =
TTYPE# = TI
TFORM# = IJ
TTNAM# = @TI
TTYPE# = EPH_ELAPSE
TFORM# = ID
TTNAM# = @EPH_ELAPSE
TTYPE# = YYYY
TFORM# = II
TNULL# = -1
TTNAM# = @YYYY
TTYPE# = MM
TFORM# = IB
TNULL# = 0
TTNAM# = @MM
TTYPE# = COL1 ...カラムのラベル
TFORM# = ID ...倍精度浮動小数点
TTNAM# = DR_A.HB_MON ...テレメトリ名称を指定
TCONV# = RAW ...生データを指定
TTYPE# = COL3
TFORM# = ID
TTNAM# = DR_A.DBL_RELOC
TCONV# = RAW
TSTAT# = @@SIB:STATUS
TSPAN# = 20

```

図5 L1TSDテンプレートの例 (FITS版)

[「レベル1時系列データフォーマット変換ツール(FITS)ユーザーズガイド」より引用]

CSV 版や SPICE CK 版のテンプレートは、テレメトリ項目名称を列挙するのみシンプルなものである。ただし、プログラムとしては、SFITSIO が使われているため、CSV 版や SPICE CK 版のテンプレートから、FITS 版のテンプレートを自動生成し動作するという作りになっている。

4.2 レコード

テレメトリから生成される時系列データには、どのようなタイミングに出力されるデータを集めてレコードを作成するかに応じ、幾つかのバリエーションがある。そこで、レベル 1 時系列データフォーマット変換ツールには、これらに対応するためのモード・機能を持たせた。

最も単純なモードは、テレメトリメッセージの種類ごとにテーブルを設け、一つのメッセージに含まれているテレメトリ項目から一つのレコードを作成するものである。このモードでは、メッセージが異なれば、これらが同じ時刻を有していても別のレコードが作成される。このモードは、作成されるファイルはコンパクトになるが、一つのテレメトリ項目が複数のテーブルに属することがあり、この場合同じテレメトリ項目でも時間により格納されるテーブルが違ってくる。

より複雑なモードは、一つでもテレメトリが得られたカラムがあればレコードを生成するものである。この場合、テレメトリが得られなかったカラムには NULL 値が設定され、値が詰まっているか否かに応じ、まだらな表が作られる。このような表が必要となる事情として、サブコマや、そもそも、種類の異なるメッセージを集めて表を作成したい場合などがある。また、あるメッセージのタイミングの値を記録したいというニーズに対応するため、図 6 に示すように指定したカラムに値があるレコードのみを残す機能を持たせた。この機能は、しばし、次節にのべるデータ補間の機能と組み合わせて使用する。

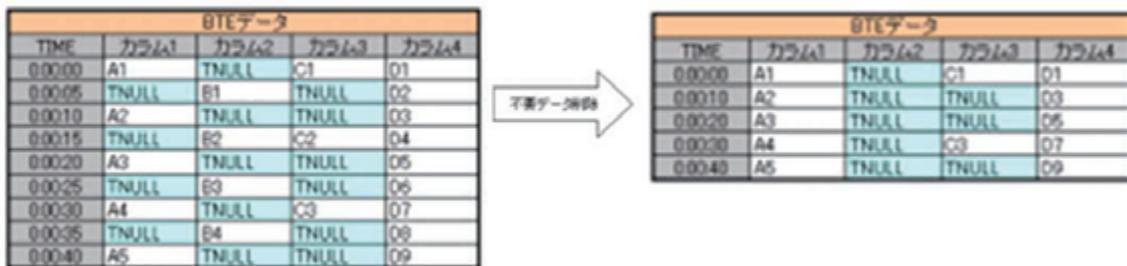


図 6 レコードの扱い

[「レベル 1 時系列データフォーマット変換ツール(FITS)ユーザーズガイド」より引用]

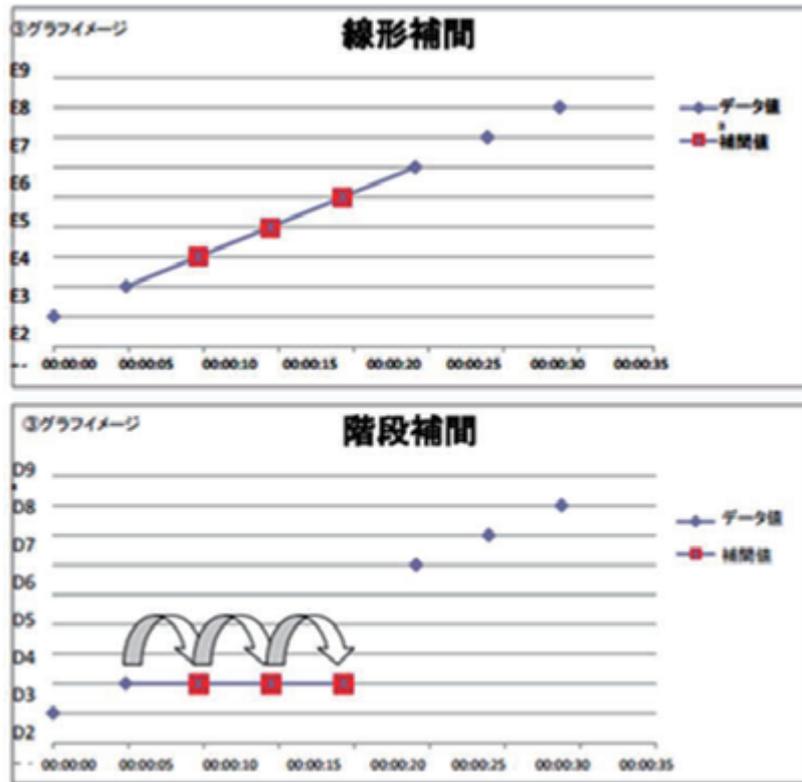


図 7 補間の方式

[「レベル 1 時系列データフォーマット変換ツール(FITS)ユーザーズガイド」 より引用]

4.3 データ補間

カラムの値に抜けがある場合でも、値を埋められるように補間の機能を実装した。この機能を用いると、前述の機能と組み合わせ、異なる出力タイミングのテレメトリ項目を集め、値の埋まった表を作成することができる。補間の方式として、線形補間と階段補間、2種類のパターンを用意した。これらのイメージを図7に示す。

サブコマの値を補間した場合など、カラムの値が補間により得られたものなのか、テレメトリで得られたものなのか、補間で得られたものなのかを知りたい場合があり得る。そこで、値が補間により得られた値か否かをビットパターンで示すカラムを作成する機能を持たせた。図8にそのイメージを示す。

(データ意味づけ)
 1: 補間済み, 0: 未補間
 (データ格納イメージ)
 例) 1レコードに3カラム存在する場合のデータ加工情報カラム

	カラム1	カラム2	カラム3	データ加工情報カラム		
0:00:00	未補間	未補間	補間済み	0	0	1
0:00:05	未補間	補間済み	未補間	0	1	0
0:00:10	補間済み	補間済み	未補間	1	1	0
0:00:15	未補間	補間済み	未補間	0	1	0

3カラム存在するのでデータ加工情報カラムは「1ビット×3カラム」付加される。

カラム1
カラム2
カラム3

図 8 補間の有無を示すカラム

[「レベル1時系列データフォーマット変換ツール(FITS)ユーザズガイド」より引用]

4.4 配布パッケージ

レベル1時系列データフォーマット変換ツールは、CSV, FITS, SPICE の3つのフォーマットに対応しているが、それぞれ異なるコミュニティが対象と想定した。そこで、それぞれのフォーマットに対して、別々の配布パッケージ・ユーザマニュアルを用意した。パッケージには、実績のあるソフトウェアを流用したため、ソースコードが開示されていないバイナリのプログラム・ライブラリも含まれている。なおバイナリは 32bit / 64bit の Linux 向けのものであるが、動作保証は Cent OS 6.5 にて行っている。また、動作には SFITSIO など、幾つかのライブラリをあらかじめインストールしておく必要がある。

5. 性能評価

表5に性能を評価した結果を示す。性能の評価には、ASTRO-Hの一次嘯みあわせ試験の総合動作確認のデータを使用した。大小のサブシステム・コンポーネントに対応し、ケース1~3を示すが、いずれも、テレメトリは同一のRPT (Raw Packet Telemetry) ファイルを読み込ませた。入力モジュールは、入力ファイルとほぼ等しいメモリを消費している。出力モジュールは、出力ファイルサイズに比べ数倍のメモリを消費しているが、出力ファイルが大きくなるにつれ倍率が小さくなる傾向がみられる。最も処理に時間を要したケース3においても、データの取得に要した時間に比べ、十分(100倍以上)に速く処理されており、実用的であることが分かる。なお、この性能測定にあたり、使用しているライブラリ SFITSIO のチューニングを実施した。チューニングの結果は、将来のリリース版に取り込まれるよう開発元に提示した。

表 5 性能評価

	ケース 1 あるコンポーネント	ケース 2 あるミッションサブシステム	ケース 3 姿勢サブシステム
入力ファイルサイズ	360M byte (実時間で約 16 時間分)		
出力ファイルサイズ	15M byte	121M byte	2,780M byte
テーブル数 / カラム数	1 / 132	93 / 6866	524 / 23162
最大使用メモリ量 (入 力 + 出力モジュール)	377M + 162M byte	377M + 1,148M byte	377M + 5,767M byte
処理時間	1m57s	2m34s	7m37s
性能評価環境	ノートパソコン : Intel Core i7-2860 QM 2.5GHz / 16Gbyte 上にて VMWare Virtual Machine を 2 プロセッサ / 16Gbyte で作成 / 64bit OS		
L1TSD のバージョン	2.05 (FITS 版)		
SFITSIO のバージョン :	1.44a に対して, 性能改善のチューニングを加えたもの		

6. 現在の使用状況

レベル 1 時系列データフォーマット変換ツールは、2012 年 3 月に初版の開発が完了し、利用可能となっている。現在、維持・改修フェーズにある。表 6 にこれまでの、ISAS の衛星プロジェクトへの適用状況を示す。これまでに、ひさき、はやぶさ 2 衛星のデータ処理に適用され、ASTRO-H のデータ処理の適用に向け、システムへの組み込みが進んでいる。これらのプロジェクトでは科学データベースの処理の一部と工学データベースのデータ処理において利用されている。これらの利用実績を踏まえ、性能の向上やエラーメッセージの分かり易さの改善など、実施の使用に基づく改善活動が進んでいる。さらに、ERG への組み込みの検討が進んでいる。

表 6 レベル 1 時系列データフォーマット変換ツールの適用状況

	CSV 版	FITS 版	SPICE 版
ひさき	○	○	○
はやぶさ 2	○	○ (予定)	○
ASTRO-H	○	○	—
ERG	○ ※1	—	—

※1: プロジェクト側が用意するプログラムにて CDF フォーマットに変換する。

7. まとめ

我々は、レベル 1 時系列データの処理を、従来に比べ少ない労力で実現するため、衛星のテレメトリの設計を記述した SIB と出力の書式を規定するテンプレートを読み込み、1 次データ処理を実現する汎用な処理プログラム、レベル 1 時系列データフォーマット変換ツールを開発した。工学データベースや、各種の学問分野の科学データベースに対応するため、CSV 版、FITS 版、SPICE 版のパッケージを用意した。これらは、実際の衛星プロジェクトへの適用が進んできている。レベル 1 時系列データフォーマット変換ツールは、オンメモリでデータ処理を行うことで高速に動作する。特に、FITS 版では、L1TSD テンプレートを作成することで、任意のデータ型を有する幅広い BTE 書式に対応できる。

参考文献

- 1) 西村佳代子, 松崎恵一, 宮澤秀幸, 高木亮治, 山下美和子, 宮野喜和, 福田盛介, 馬場肇, 永松弘行, 山田隆弘: 「SIB2/GSTOS-1 における開発状況」, 宇宙科学情報解析論文誌 第三号 (2013), p17. <http://repository.tksc.jaxa.jp/pl/dr/AA0062302003>
- 2) 高木亮治, 北條勝己: 「EDISON(衛星運用工学データベースシステム)の開発」, 宇宙科学情報解析論文誌 第五号 (2015), submitted
- 3) Tamura, T., Baba, H, Matsuzaki, K, Miura, A, Shinohara, I., Nagase, F., Fukushi, M., Uchida, K. - “Data Archive and Transfer System (DARTS) of ISAS”, Astronomical Data Analysis Software and Systems (ADASS) XIII, Proceedings of the conference held 12-15 October, 2003 in Strasbourg, France. Ed. Ochsenbein, F., Allen, M. G., and Egret, D.. ASP Conference Proceedings, Vol. 314. San Francisco: Astronomical Society of the Pacific, (2004)., p22.
- 4) Wells, D. C., Greisen, E. W., and Harten, R. H., “FITS: A Flexible Image Transport System”, Astron. Astrophysics. Supplement, 44 (1981), p363.
- 5) “CDF Documentation”, http://cdf.gsfc.nasa.gov/html/CDF_docs.html
- 6) Acton, C. H., “SPICE - An Observation Geometry System for Planetary Science Missions”, Planetary and Space Science, Vol. 44, Issue 1, (1996), p65.
- 7) FITS の手引き 第 5.3 版, 国立天文台 天文データセンター 発行, 2013 年 1 月 24 日
- 8) McMahon, S. K., “Overview of the Planetary Data System”, Planetary and Space Science, Vol. 44, Issue 1, (1996), p3.
- 9) Yamauchi, C., “SFITSIO - A Next-Generation FITS I/O Library for C/C++ Users”, Astronomical Data Analysis Software and Systems XIX. Proceedings of a conference held October 4-8, 2009 in Sapporo, Japan. Ed. Mizumoto, Y., Morita, K., and Ohishi, M.. ASP Conference Series, Vol. 434. San Francisco: Astronomical Society of the Pacific, (2010), p.469.
- 10) Pence, W., “CFITSIO, v2.0: A New Full-Featured Data Interface”, Astronomical Data Analysis Software and Systems VIII, ASP Conference Series, Vol. 172. Ed. Mehringer, D. M., Plante, R. L., and Roberts, D. A., (1999), p. 487.

謝辞

レベル 1 時系列データフォーマット変換ツールは, 日本電気株式会社, 日本電気航空宇宙システム株式会社の協力の下, ソフトウェアの開発を実施しました. また, 開発にあたり, SFITSIO 開発者の山内千里様の全面的な支援を受けるとともに, 幾つかの機能提供を受けました. また, ASTRO-H プロジェクトの皆様に, 本ツールに対する様々なフィードバックをいただきました. 構想の具現化に協力を頂いた方々に, 感謝の意を表します.

超小型位置天文衛星 Nano-JASMINE のデータ解析

山田 良透^{*1}

Data analysis of Ultra small astrometry satellite Nano-JASMINE raw data

Yoshiyuki YAMADA^{*1}

ABSTRACT

Nano-JASMINE is high precision astrometric satellite mission by using ultra small satellite which have developed rapidly in recent decades. Targeted accuracy is 3 milliarcsec (3 mas), or 1.5×10^{-8} rad. For example, attitude sensor of middle sized satellite has about 1 arcsecond accuracy. On the other hand, that of ultra-small satellite has 1 arcmin accuracy. The key of achieving such high accurate observation by poor satellite instruments is the role of “data analysis”. In this document, we will report Nano-JASMINE data analysis, especially parameter selection of attitude model and its validation.

Keywords: Nano-JASMINE, Data analysis, Model

概 要

Nano-JASMINE は、近年急速に進歩している超小型衛星を利用して、高精度な星の位置決定の観測をしようという試みである。Nano-JASMINEの目指す精度は3ミリ秒角(3mas), 1.5×10^{-8} radである。衛星の姿勢センサーをみると、大型衛星では1秒角(1/3600度)程度が達成されるのに対して、超小型衛星では1分角(1/60度)程度の姿勢決定精度しか持たない。大型衛星に比べると性能の低い機器を用いた衛星で、高精度観測を達成するカギは、データ解析の役割の重要性である。Nano-JASMINEにおけるデータ解析、特に姿勢モデルのパラメータの選択に関して報告する。

記 号

mas	ミリ秒角	\mathbf{b}, \vec{b}	独立パラメータ
μas	マイクロ秒角	$f_i(\vec{a}, \vec{b})$	i 番目の観測値を決める式
HILS	hardware in the loop simulator	x_i	i 番目の観測期待値
CCD	Charge coupled device	o_i	i 番目の観測値
TDI	Time delayed integration	ε_i	i 番目の観測誤差
i	観測の index	\mathbf{q}_j	j 番目の姿勢 quaternion
j	姿勢時系列の index		
\mathbf{a}, \vec{a}	最小二乗法における推定パラメータ		

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

^{*1} 京都大学大学院理学研究科 (Graduate School of Science, Kyoto University)

1. はじめに

日本の国立天文台を中心とするグループでは、超小型衛星を利用した Nano-JASMINE という位置天文衛星の打ち上げを予定している¹⁾。位置天文衛星と言えば、ESA により打ち上げられた Hipparcos(1989~1993)があり、同じく ESA により打ち上げられた Gaia(2013~)が運用中である。Hipparcos 衛星は、重量 1.4t の衛星だったが、Nano-JASMINE はたった 35kg の衛星で、Hipparcos と同程度の精度のカタログを出すことを目標としている。なぜそのようなことが可能になるかと言うと、もちろん、衛星バス部の搭載機器である Reaction Wheel や通信機、オンボード処理装置などが小型化したという事情もある。また、Hipparcos は光電増倍管を使っていたが、今は宇宙で CCD を使うことができるようになったという事情もある。位置天文衛星では解析の占める役割が大きいことが、小さな衛星で高精度を達成できる理由である。本稿では、Nano-JASMINE 衛星のデータ解析について、なぜ 3mas の精度が達成できるのかに係る解析技術を中心に、紹介する。

2. Nano-JASMINE 計画

位置天文観測は、星の時々刻々の位置を測定し、これより星の天球上の運動を精密に測定する天文学である。地球が太陽の周りを公転することにより、星は天球上で小さな楕円運動をする。また、星は銀河系の中を運動するが、連星など特別の星を除けば加速度は観測にかかるよりはるかに小さいので、星の運動は銀河系に固定した座標で見れば等速直線運動だと思って良い。地球、あるいは地球に近い軌道を周回する衛星から星の運動を観測すると、この見かけの楕円運動と等速直線運動の合成運動として、星は天球上をらせん運動する。天球面上の二次元等速直線運動はある基準時刻での位置と速度の合計 4 つのパラメータで、また楕円運動の位相と長半径・短半径比は星のある方向で決まるので、楕円運動は長半径だけ表現できる。これら 5 つパラメータを、位置天文パラメータと呼び、これらを解くことが位置天文の目的である。

楕円運動の長半径を年周視差と呼んで、これは太陽系重心から対象とする恒星までの距離の逆数である。天文学において、距離を測定することは、見かけの明るさから実際のエネルギーに換算する上で、また恒星系の 3 次元的な位置関係を把握するために重要である。しかし、多くは主系列星の色と等級の関係を仮定したり、変光星の周期と光度の関係を仮定したり、様々な仮定にもとづいている。地球の力学的な運動だけから推定される星の運動をもとに、それ以外の特別な仮定無しに距離を測定することができるが、位置天文学の特徴である。

ところが、年周視差と距離が逆数関係にあることや、星が遠くほどたくさん存在することから、年周視差の誤差が 10% を越えると距離の誤差は急速に増大する²⁾。距離 100pc に対応する年周視差は 10mas(ミリ秒角)だが、Hipparcos の精度である 1mas はこの 10% に相当する。つまり、今のところ直接測定で測られている星の距離は 100pc までと言うことである。Gaia 衛星は 10 μ as の精度を達成する予定なので、年周視差 100 μ as に相当する 10kpc 程度の星の距離まで測定できることになる。

Nano-JASMINE は、わずか口径 5cm の望遠鏡で、Hipparcos が達成した 1mas レベルの位置天文精度をもつカタログを作るためのミッションである。衛星バスは東京大学工学部、ミッション機器は国立天文台と京都大学、打ち上げはウクライナのロケットでブラジルの射場から行われるという、大きな国際協力である。さらに、データ解析は、日本のチームとヨーロッパの大型衛星 Gaia の解析チームが共同で行っていて、ドイツ、フランス、スペイン、オランダ、イギリス、ロシア、スウェーデン、アイルランドなど多くの国の研究者がかかわっている。特に Gaia のチームは Hipparcos で、位置天文のむずかしさとそれに対する対処の経験を持ち合わせている。

Hipparcos の最大の教訓は、次の点である。通常システムエンジニアリングでは、プロジェクトの成

功の前提にはミッションの成功が、ミッションの成功の前提にはシステムの成功があると考えられている。しかし、Hipparcos は予定の軌道に投入できなかった。明らかにシステムとしては失敗のミッションであるが、天文学的には非常に大きな成果を残した。プロジェクトとしては成功である。これを支えていたのが、解析技術である。予想していた軌道をそれでも、衛星から得られる様々な情報をもとに、衛星のモデルを組み直し、新しいモデルのもとで解析を実施し、当初予定以上の成果を出すことに成功した。

Nano-JASMINE 衛星の仕様は表の通りである。衛星自身は 2010 年 10 月に組みあがっていて、現在は東京大学工学部のクリーンブース内に保管されている。Nano-JASMINE は Hipparcos と同様、スピンの軸に垂直な方向の大角度離れた 2 視野を同時に観測する。2 視野の相対角は、Nano-JASMINE の場合は 99.5° である。Hipparcos では約 60° 、Gaia では約 106° である。二視野を同時に観測するため、主鏡の前にビーム混合鏡と呼ばれる確度の付いた二枚の平面鏡を置き、主鏡の上半分と下半分のそれぞれが別の方向を見るように設計されている。スピン周期は Nano-JASMINE で 1.5 時間、Gaia では 6 時間程度だが、このスピン数周回中で二視野の相対角が安定していることが、高い位置精度を達成するカギとなる。

表 1 Nano-JASMINE の仕様概要

質量	35kg
大きさ	50cm 立方
姿勢制御	三軸安定
ミッション期間	2 年
軌道	太陽同期軌道(高度 800km), LTAN 22 時
打ち上げ	2017 年末(調整中)
主鏡口径	5.3cm
2 視野の相対角	99.5°
焦点距離	1.67m
視野	$0.5^\circ \times 0.56^\circ$
検出器	完全空乏裏面照射型 CCD 1024 pixel \times 1152pixel
ピクセルサイズ	$15\mu\text{m} \times 15\mu\text{m}$, 1.8 秒角 \times 1.8 秒角
自転周期	100 分(軌道周期と同期)
観測波長帯	$0.6\mu\text{m} \sim 1.0\mu\text{m}$ (z_w バンド)
精度	$\sim 3\text{mas}$ @ $z_w=7.5$ (Vega 等級)

また、衛星を熱的に安定させるため、環境条件をできるだけ均一にしなければならない。スピンに垂直方向の二つの視野で全天を掃くには、スピン軸を動かさなければいけないが、太陽からの角度を一定角度の 45° に保ったまま、2 か月程度の周期で歳差させる。スピン軸方向が太陽指向方向と一定であるということは、熱的に安定した環境を得るための一つの工夫である。CCD を TDI 駆動することで、両視野は天球上を掃きながら連続的な帯状のデータを出してくる。発生データレートは 2Mbps 程度となる。地球周回衛星では地上のアンテナからの可視時間は全体の 5% 程度なので、通常の科学衛星の通信では全データはダウンリンクできない。そこで、星像中心のまわりの $5 \times 9\text{pixel}$ 程度の部分を切り取って、データをダウンリンクすることで、2Mbps のダウンリンクレートでもダウンリンク可能なデータ量に減らすことができる。

3. Nano-JASMINE データ解析とモデル

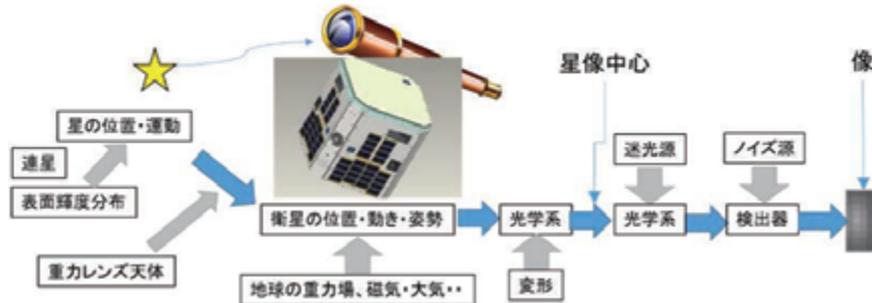


図 1 位置天文観測データの解析において考慮すべき効果。

Nano-JASMINE のデータ解析は、地上に降ろされるデータから星の位置や運動を再構築することである。これには、図 1 のような流れの逆問題を解く。図 1 を説明すると、次のようになる。

最初に星の位置と運動、衛星の姿勢や光学系のゆがみなどの情報がすべてわかっていたとする。そうすると、像面での星の中心位置は容易に計算できる。ただし、これらの効果の取り入れ方は、慎重に行わなければならない。

我々が測定しているのは星の像の光学中心である。一方、星が銀河系内を等速直線運動するのは、重力中心の話である。もし光学中心と重力中心が異なる動きをすれば、星の運動が 5 つのパラメータで書けるという前提が間違いになる。さらには、赤色巨星のような星はその半径が 1AU 程度ある。その星が黒点を持っていたり、光度にムラがあったりすると、光学中心の運動も観測精度の範囲で星の運動モデルからのずれを生じる。星の光が望遠鏡に入る前に、重力レンズ効果を受けて、星のある方向と光の来る方向がずれることも考えられる。望遠鏡の指向が正しく表現されたとしても、光学系がゆがんでいれば、像面の位置と天球上の位置の写像が自明ではない。さらに、衛星が運動しているので光行差も考慮しなければならない。

図 1 中の四角で囲ったすべての項目は、天球上の星の位置から像面での星像中心、さらには像を生成するときに影響を与える項目であり、仮にそれらが全てわかっていたとすれば、これは簡単な数式で書ける。

$$f_i(\vec{a}, \vec{b}) = x_i, o_i = x_i + \varepsilon_i \quad (1)$$

式中で f がその写像、 a は、図 1 でリストアップした「全ての」効果のモデルのパラメータ、 b は時刻などの独立変数である。観測を index i で表すと、 x_i は与えられたパラメータでの i 番目の測定での星像中心の期待値を表す。観測値 o_i は、 x_i に対して誤差が加わったものである。像面での星の中心位置から像を計算するのも同様のプロセスである。光学系のゆがみ、検出器の特性やノイズ、迷光などの条件を与えれば計算はできる。

位置天文解析は、与えられた o から a を逆問題として解くものである。Hipparcos でも Nano-JASMINE でも Gaia でも、この逆問題を、最小二乗法を用いて解く。

ここで重要な点は、最小二乗法とは

- 解きたい物理現象(姿勢・軌道・星の運動)を指定しても問題は定まらない。
- 物理現象を表現する「モデル」を指定して、初めて問題が定まる。

ということである。位置天文解析では、星の運動は先に述べた 5 パラメータの運動をするという「モデル」をたてる。Nano-JASMINE では、衛星姿勢は姿勢 quaternion q_j を区分的 Spline でつないだモデルを採用し、軌道はチェビシェフ多項式、星像は PSF のモデルを光学計算し、像面の変形、放射線の影響、chromaticity などは適当な次数の多項式を用いる。

4. Nano-JASMINE の姿勢モデル

Hipparcos のデータは 1997 年に ESA より公式に公開されたが、その 10 年後に新しい解析を行った結果が van Leeuwen 氏によって公表された³⁾。この研究から、まったく同じ生データでも、モデルを改善すると精度が改善することである。特に重要だと指摘された点は、衛星姿勢のモデルである。

Hipparcos では、スピン垂直方向の二つの視野を星が通過する時刻を測定する。つまり、スキャン方向(スピン軸に対する経度方向)の座標だけを測定する。スピン軸方向の評価に誤差があったとしても、スキャン方向座標は軸の方向の誤差に対して二次でしか効かないので、結果に対するインパクトは少ないというのがシステム設計上のポイントであった。しかし、これはスピン軸方向の評価に多少の誤差があるとしても、軸方向そのものは安定しているという「姿勢モデル」の前提がある。実際のデータを見ると、この前提は必ずしも正しくないというのが、van Leeuwen 氏の指摘である³⁾。結果として、スキャン方向座標だけを精度よく測っても、軸の動きを含めたモデル化をしないと、十分な精度は達成できないということである。van Leeuwen 氏は、衛星姿勢に対する様々な影響を物理的に考慮して、姿勢 quaternion を区分的 spline でモデル化することを提案した。

Nano-JASMINE 衛星は、Hipparcos 衛星に比べて軽く、軌道上で様々な擾乱を受けやすい。そのため、姿勢モデルについては Hipparcos 衛星以上に慎重に考える必要がある。システム同定モデルを適用すると、残差が白色化できる⁴⁾という研究結果もある。しかしながら、位置天文解析は大規模な解析であり、モデルを大幅に変更することは難しい。我々は、星像中心の像面上の座標から位置天文パラメータを用いる部分は、Gaia のコアソフトウェア AGIS⁵⁾ を適用する予定である。このソフトウェアは、姿勢モデルとして姿勢 quaternion の区分的 spline モデルを採用している。

このモデルを大幅に変更できないという前提で、可能なパラメータの中で必要な精度の姿勢 fitting を行うことができるかどうかについて、検討した。姿勢データとしては、東京大学工学部の Nano-JASMINE 担当学生が開発した HILS (Hardware in the loop simulator) を用いた。これには、地磁気、太陽輻射、大気などの姿勢擾乱要因、センサーの精度に伴う姿勢センシング誤差と、そのセンサー情報をもとにした姿勢制御則が実装されている。

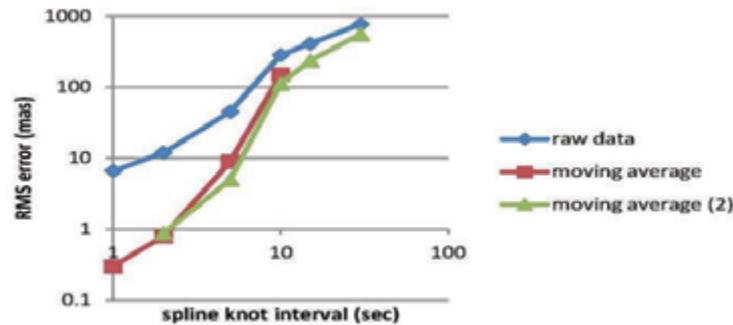


図 2 HILS により計算された姿勢則を Spline モデルで fit した場合の残差。青は生の HILS データに対するもの、赤と緑は HILS データに対して撮像時間の移動平均をとったものである。

HILS により計算される姿勢に対して、knot 間隔が異なるいくつかの Spline モデルで fit し、fit 前の姿勢との残差の RMS 値を計算したのが、図 2 のグラフである。位置天文解析のデータに使われる星像は、撮像時間程度(10 秒)の積分値である。それ以下の細かな振動は、星像中心を求める際は、その積分値だけが問題となる。そこで、生のデータへの fitting のほかに、10 秒移動平均をとったものに対する fitting も行った。移動平均の二つの異なる線があるのは、一つのデータで短時間スケールから長時間スケールまでをカバーできなかったため、二回の異なるシミュレーション結果を使ったためである。

Nano-JASMINE の要求精度は位置天文パラメータの精度で 3mas である。姿勢 fitting の誤差は、ある種の誤差として効くことになるが、異なる観測での姿勢相関は無いことから、本当はこれを観測回数 N_0 に対して $1/\sqrt{N_0}$ 倍程度に低減することを期待しても良いという考え方もあり得る。この姿勢誤差が「統計誤差」だとして扱う方法である。一方で、残差の白色性は保証されていないことを考えると、「系統誤差」として扱うべきだという考え方もあり得る。このどちらの考え方が正しいのか、あるいはその中間のどのあたりに答えがあるのかは、end to end のシミュレーションを実施してみないと分からない。そこで、今回は安全側に考えて、系統誤差的な取り扱い、即ち fitting 残差が 3mas より十分小さいことを条件とする。そうすると、knot 間隔が約 3 秒以下であれば、Spline モデルでも fit できるという結論になる。

ここでは、Nano-JASMINE の姿勢モデルのパラメータについて評価を行った。同様に、軌道、光学系ゆがみ、検出器特性、PSF など、Nano-JASMINE の観測に関係するあらゆる物理量のモデルを選択し、要求精度を達成するために必要な次数やパラメータ数を評価し、これらを解析ソフトに実装する必要がある。地上での実証実験などをもとに、これらの評価を進めている。

5. 結論

我々は、超小型衛星による位置天文観測衛星 Nano-JASMINE のデータ解析の準備を進めている。観測ストラテジが同じ Gaia 衛星のデータ解析チームと、コアソフトウェアの部分の共有を行う予定である。同じソフトウェアを異なるミッションに適用することは、チャレンジングである。その中で、解析に使われる「モデル」が取り得るパラメータを適切に選んで、系統誤差要因が目標精度の 3mas に比べて十分小さいようなパラメータが取れるかどうかの検討を進める。そのため、大学衛星の強みを生かし、工学・理学が一体となって、モデルの改善に努めている。さらに、モデルのより柔軟な変更が可能になるフレームワークも開発中である⁶⁾。

参考文献

- 1) Gouda, N., "JASMINE", Scholarpedia, 6(10), 2011, 12021.
- 2) Lutz, T. E. and Kelker, D.H., On the Use of Trigonometric Parallaxes for the Calibration of Luminosity Systems: Theory, Publ. Astron. Soc. Pacific, 85 No.507,1973, pp.573- 578
- 3) van Leeuwen, F., "Hipparcos, the New Reduction of the Raw Data", Springer, 2007
- 4) 伊藤まゆ美, "超小型人工衛星を用いたシステム同定による位置天文観測衛星に関する基礎研究", 慶應義塾大学大学院理工学研究科修士論文 (2013)
- 5) Lindegren, L., Lammers, U., Hobbs, D., O'Mullane, W., Bastian, U., and Hernández, J., "The astrometric core solution for the Gaia mission. Overview of models, algorithms, and software implementation", Astronomy and Astrophysics, 538 (2012) pp.78L
- 6) 初鳥陽一, 宮下尚, 清水淳也, 山田良透, "モデル駆動型システムズエンジニアリングに基づくモデル管理およびデータ解析—Nano-JASMINE データ解析への応用—", 宇宙科学情報解析論文誌, 2 JAXA-RR-12-006 (2013), pp.103-111.

月周回衛星「かぐや」のデータを用いた 多目的最適化による月着陸最適候補地の選定

西山 万里^{*1}, 大嶽 久志^{*2}, 星野 健^{*3}, 橋本 樹明^{*2},
渡辺 毅^{*2}, 立川 智章^{*4}, 大山 聖^{*2}

Selection of Landing Sites for Lunar Lander with “KAGUYA” Data Using Multi-Objective Optimization

Mari NISHIYAMA^{*1}, Hisashi OTAKE^{*2}, Takeshi HOSHINO^{*3}, Tatsuaki HASHIMOTO^{*2},
Takeshi WATANABE^{*2}, Tomoaki TATSUAKI^{*4}, Akira OYAMA^{*2}

Abstract

The Japanese lunar orbiter spacecraft “KAGUYA” had obtained enormous scientific data about the moon using equipped cameras and sensors. In this research, we created a moon database with a moon simulator that developed using the moon data, such as illuminated time, communicable time, inclination angles and ice distribution. We analyzed those data for selecting landing points suitable for lunar missions using Multi-Objective Optimization. This paper describes the method how we selected landing points in terms of both technical and scientific aspects, and the obtained knowledge we derived from the results of this research.

Keywords : Lunar Polar Expoloration, Landing Site Selection, Multi-objective Optimization

概要

月周回衛星「かぐや」は、搭載された地形カメラをはじめとする各種センサを用い、月に関する膨大なデータを収集した。本研究では、これらのデータを元に開発された月面における日照、通信、傾斜角をシミュレーションするソフトを用いて年間の月南極付近のデータベースを作成し、また報告されている月の氷の存在確率に関する情報を加えて宇宙探査における技術・ミッション両方の観点から解析することで、月探査ミッションにおける多目的最適な着陸候補地点の選定を行い、将来の月探査ミッションにおいて有用な知見を導き出した。

1. はじめに

人類はこれまで数多くの月探査を行ってきた。古くは1959年に打ち上げられたルナ1号に始まり、アメリカのアポロ計画では有人月探査が行われたことは、世界的に広く知られていることである。日本でも1990年代から月周回衛星「かぐや」による月の詳細な観測が行われ、数多くの科学的なデータが取得された。このデータを用いて次期月探査計画のための着陸地点検討が行われているが、これまでは大域的に日照条件の良い地域の中から人間が選んだ範囲をより詳細に分析することを繰り返して候補地点を挙げるなど、広範囲を定量的に分析するような着陸地点の選定は行われていなかった。これに対し本研究では、探索範囲を広げながらも制約条件によって探索点が膨大なることを防ぎ、さらに多目的最適化を用いることで着陸地点に対する技術的・ミッション的要求を満たすような候補地点を選定することを目指した。

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

^{*1} 東京大学大学院工学系研究科

^{*2} 宇宙航空研究開発機構宇宙科学研究所

^{*3} 宇宙航空研究開発機構宇宙探査イノベーションハブ

^{*4} 東京理科大学工学部第一部経営工学科

本論文の構成は以下の通りである。まず第2章では、月面探査機の着陸地点の選定、および多目的最適化に関する背景を述べる。第3章では、多目的最適化を用いた着陸地点の探索手法について詳細を述べる。第4章では得られた結果について述べ、そこから導き出される知見について考察する。最後に第5章で、本論文のまとめを行い、今後の課題について述べる。

2. 背景

2.1. 月面探査機の着陸地点に対する要求

月探査計画において、月面探査機の着陸地点は非常に重要な意味を持つ。たとえば月面上における電力供給は太陽光発電に依存するため、日照の確保は死活問題である。また、単に着陸するだけでなくその後の月面探査を見据えた場合、科学的に興味深い物質や地形の存在する地点の近辺に着陸することが望ましい。そのため、探査機の着陸地点は様々な条件を加味し、注意深く選定される必要がある。

ここで、月面探査機の着陸地点に対しては技術的要求とミッション的要求の二種類が存在する。技術的要求とは探査機が機能を維持するために必要な要求であり、例として以下の3点が挙げられる。

- A) 日照条件の良さ
- B) 地上との通信条件の良さ
- C) 傾斜角の小ささ

Aの日照条件は太陽光発電による発電に影響するため、探査機の機能維持を大きく左右する。日陰の時間が短くなるほど蓄電に必要なバッテリーの容量を小さくすることが出来るため、連続して続く日陰時間が短いほど望ましい。またBの通信条件については、観測データやコマンドを地上との間で送受信するために地球が可視である期間がより長いことが望ましい。さらにCの傾斜角についてだが、着陸地点の傾斜があまりに急である場合、着陸機が転倒し計画全体が失敗に終わってしまう危険性が存在する。さらに着陸後探査機からローバを降ろすこと、このローバが周辺を走行し探査を行うことも考慮すると、傾斜角はなるべく小さい方が望ましい。

一方ミッション的要求とは探査をより有意義なものとするための要求であり、科学的な観測データが得られることが期待できるような条件のことを示す。例としては着陸地点付近における氷の存在確率の高さや、月の起源に関わる物質を含んだ鉱物の多く存在する地点への近さなどが挙げられる。

これら複数の目的を満たすような着陸地点を選定することによって、計画を遂行しやすく、なおかつ科学的成果をより多く得られるような探査を実現することが可能となる。しかしながらこれらの目的は必ずしも両立するものではなく、時に相反することもある。たとえば日当たりの良いところでは氷が溶解してしまうため、日照条件の良さと氷の存在可能性は両立することが難しいなど、トレードオフの関係にある目的の組はいくつも存在する。

こうした複数の目的を扱うにあたり、本研究では多目的最適化を用いることを提案した。以下では多目的最適化の特徴、および従来の単目的最適化との違いについて述べる。

2.2. 多目的最適化

工学の様々な分野において、複数の目的を有する最適化問題は数多く存在する。これらの目的は一方の改善により他方も改善する関係であることもあるが、一方の目的を満足するために他方が犠牲になる、すなわち改悪されるケースもしばしば存在する。そのため、複数の目的間で妥協解を探索することが必要となってくる。

こうした相反する多目的問題を扱う場合、従来は単目的最適化（スカラー化）手法が広く用いられてきた。これは各目的関数に対し重み付けを行い1つの評価式で表すことで、多目的問題を単目的問題とみなす方法である。しかしながらこの手法では、各目的関数の重み付け係数の定め方は経験や観測結果などに基づくことが多く、不確かさを有していた。さらに1度の試行において1つの最適解しか求められないため、複数の解を発見したい場合には何度も試行を繰り返さねばならなかった。

これに対し、多目的最適化問題の解を同時に複数求める多目的最適化手法が存在する。多目的最適化では各目的関数を個別に評価するため、距離と量など異なる単位や尺度を持った目的関数値を扱うことも可能となる。このような多目的最適化により導き出された最適解は「パレート最適解」と呼ばれ、ある目的関数の値を改善しようとする、他の目的関数の値が悪化してしまうような解である。自身よりも勝る解が存在するような劣解に対し、他のどの解よりも劣らないという点からパレート最適解を非劣解と呼ぶこともある。単目的最適化と多目的最適化の比較を図1に示す。

多目的最適解の評価は Pareto Ranking により行っている。これは Fonseca らにより提唱された手法[1]であり、各解において自身より全ての目的関数値において勝る解の個数によりランクを定め、最高ランクのものをパレート最適解として定義する(図2)。これらのパレート最適解を繋いだ曲線がパレート曲線と呼ばれ、この曲線上に最適解が分布している。

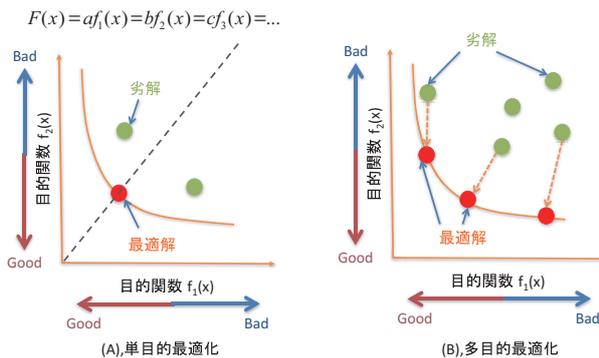


図1 単目的最適化と多目的最適化の比較

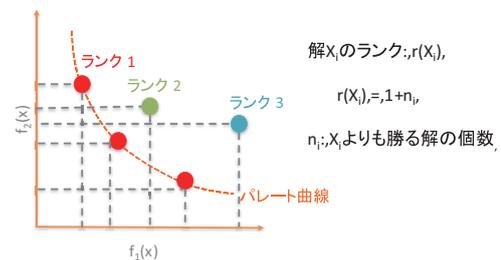


図2 Pareto Ranking

3. 多目的最適化による着陸地点探索手法

これまで、宇宙探査における最適化問題に多目的最適化を用いた例は少なく、適用される対象も軌道設計や構造設計などの分野に留まっていた[2][3]。本研究ではこの手法を着陸地点の選定に用いることで、従来発見されていなかった複数の目的に対し良好な着陸候補地点を見付け出すことを目的としている。

3.1. 着陸範囲の選定

月面探査機の着陸地点を検討するにあたり、まず選定範囲を月の南極点周辺とした。この範囲を選定した理由として、第一に日照条件が挙げられる。月は公転周期と自転周期がほぼ一致しており、そのため中低緯度の範囲においては2週間の昼と2週間の夜が繰り返して到来する。夜の間には日照が得られないため太陽電池を使用することが出来ず、蓄電により電力を賄おうとする場合、100Wあたりで300kg近い電池が必要となる。一方、月の極域であれば長期間の日照を得られるため、太陽電池の定期的な電力供給が期待できる[4]。

第二の理由として、温度環境が挙げられる。中低緯度の範囲では、昼の間の最低表面温度が120°Cに達し、放熱を行うのがきわめて困難となる。また夜の間は最低表面温度が-200°Cまで低下し、これに堪えるだけの保温・断熱機能を有する必要がある。一方で極域における表面温度はほぼ-50±10°Cと安定しており、中低緯度の地域に比べ機能維持の点で望ましいと考えられる[4]。

第三の理由として、月の南極周辺には科学的に興味深い地形が多く存在していることが挙げられる。たとえば月の南極に存在するシャクルトンクレーターの底には、年間を通じて太陽光の当たらない永久影が存在し、1990年代にアメリカの月探査機クレメンタインおよびルナ・プロスペクターの行った観測により、氷の存在が示唆されていた[5][6]。その後日本の月周回衛星「かぐや」の地形カメラにより10m精度の高解像度データが取得され、クレーターの底表面部に氷は存在しないことが確認されているが、土やレゴリスに混ざっている可能性はなおも残されている。

以上の理由から、本研究では月の南極点を中心とする四方300kmの範囲を探索の対象と定めた。

3.2. 月面データベースの作成

各着陸地点候補の評価において必要となるデータを収集するため、本研究では NTT データ CCS により作成された月面シミュレーションソフトウェアを使用した。本シミュレーションソフトウェアは指定した地点およびその周囲の日照の有無、地上との通信の可否、およびその地点の傾斜角を算出することが出来る。

日照については太陽を1点の物体として扱い、Ray Tracing 法により太陽光線と地点との間に遮蔽物が存在しないかどうかをシミュレーションしている。なお、常に太陽高度が低く方位角が全周であるという特徴から、月極域探査機では円筒状の太陽電池パネルを垂直に立てて発電を行うことが予想される。そのため発電量は太陽光の入射角よりも、月の地形に遮られず到達する太陽光の割合に依存する。よって本研究では日照量の大小は区別せず、少しでも日照があれば1、日照が皆無であれば0として扱い、計算時間の短縮を試みている。通信についても同様に Ray Tracing 法を用いており、日本の地上局に限らず地球が可視であれば通信可能と定義されている。なお、地上からの高さが5mである地点の計算を行っている。これは探査機の構造上、太陽電池パネルや通信機器が設置される位置が最大で5m前後であるという想定に基づいている。このシミュレーションソフトウェアを用い、月の南極点の四方300kmにおいて10m精度で2019年1月1日から2019年12月31日までのデータを1日おきに取得した。これらの全情報を月面データベースとしてまとめたものを、以降の地点評価および探索において用いている。

3.3. 制約条件による着陸候補地点の制限

着陸地点には月面着陸機の工学的な仕様により、以下の2つの制約条件が存在する。

1. 着陸地点の傾斜角は15°より小さくなければならない。
2. 着陸地点において連続的に続く夜の長さは14日より短くなければならない。

1の制約条件は、探査機が現実的なリソースで転倒せず着陸可能な傾斜角の目安である。2の制約条件は、現実的なバッテリーの容量規模を考慮し14日を上限とした。ここで14日という値は、夜が14日間続く月の中低緯度に対し、より夜の期間が短い場所の多い月極域での探査を行う上での利点を活かすことを目的として設定している。本研究では、これら2つの制約条件をとともに満たす着陸候補地点のみを実行可能解として扱うこととした。

3.2で述べた月面データベースに対し、以上の2制約条件を元に月の南極点の四方300kmの範囲について10m単位のグリッドサーチにより着陸候補地点の制限を行ったところ、図3に示す通り実行可能解として残ったのは174193点であった。当初は四方300kmを10m精度で全探索するため9億点の候補が存在していたが、制約条件を導入することにより探索点の個数は全探索時の0.019%まで絞られ、探索の大幅な高速化に寄与することとなった。

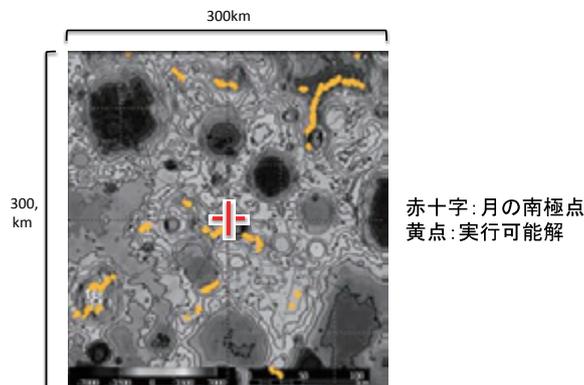


図3 制約条件により導かれた実行可能解

3.4. 目的関数の設定

本研究において、着陸地点を評価する目的関数として設定したのは以下の4目的である。いずれの目的関数値も小さくなるほど良く、0から1の範囲に収まるよう設定されている。なお、本研究では着陸日を任意に選ぶことが出来ると仮定している。

I. 日照条件の良さ（着陸時期により変動するため時変量）

日照条件の評価においては、日照そのものではなく夜が連続して続く日数を用いている。年間の最大連続夜日数を、制約条件の14日で割った値を目的関数値とする。なおバッテリーの充電は急速に行うことができるため、充電に必要な日照期間ではなくバッテリー容量に影響する日陰期間を目的関数として用いた。

II. 通信条件の良さ（着陸時期により変動するため時変量）

通信条件の評価においては、通信可能かつ日照の存在する日数を用いている。これは通信機器の作動に電力を多く要するため、充電可能な状況下においてのみ使用可能となるためである。年間を通した通信可能かつ日照の存在する日数の累計を365日で割り、それを1から引いた値を目的関数値とする。

III. 傾斜角の小ささ（地点固有の値のため固定量）

傾斜角の評価においては、その地点の傾斜角を制約条件である15°で割った値を目的関数値とする。

IV. 氷の存在確率（地点固有の値のため固定量）

月面における氷の存在確率については、着陸候補地点から半径1km以内に存在する最も近い氷までの3次元距離に氷の深さを掛け合わせたものを目的関数値としている。なお、存在する氷の分布についてはPaigeらがDivinerを用いて観測を行った結果[7]を元としている（図4）。さらに3次元距離に関しては、かぐやにより観測された月の高度データを用いている。

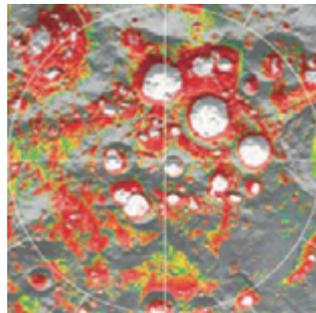


図4 月の南極域に存在すると考えられる氷の分布

以上の4目的に関して、制約条件を満たした全実行可能解について各目的関数値を求め、それらに対しPareto Rankingを用いることで多目的最適解を探索した。設計変数として扱ったのは月面をポーラステレオ投影法により表した図3の南極点を中心として図を分割し、各図の左上の点を(0, 0)、右下の点を(15001, 15001)と定義したデカルト座標系のx座標およびy座標であり、計算環境は以下の表1に示す通りである。南極点を原点と見立てて4分割したそれぞれの範囲を第1象限から第4象限と定義し、12コア全てを用い各象限の計算を並行して行った。各象限において制約条件を満たす地点の目的関数の計算に掛かった実行時間、およびこれが制約条件に縛られない全探索であった場合の予想実行時間を表2に示す。なお、

$$(\text{予想実行時間}) = (\text{実際に掛かった実行時間}) \times (\text{全地点数}) / (\text{実行可能解数})$$

として計算を行っている。表2から分かる通り、何の制約も設けず全探索を行うと計算に5,6年近い期間を要するため現実的ではなく、制約条件による絞り込みが必要不可欠となってくる。

表1 計算環境

CPU	Intel(R) Xeon(R) CPU E5-2697 v2 (2.70GHz)
OS	Fedora 20
メモリ	128GB
コア数	12 cores

表2 実行時間

象限	1	2	3	4
実行可能解数	63395 点	20718 点	76674 点	13406 点
実際の実行時間	12 時間 17 分 18 秒	4 時間 57 分 15 秒	14 時間 52 分 27 秒	3 時間 18 分 40 秒
全探索予想実行時間	約 1816 日	約 2240 日	約 1818 日	約 2308 日

4. 結果および考察

探索の結果、得られた多目的最適な着陸地点候補は 17413 点存在した。各地点の分布は図5に示す通りである。

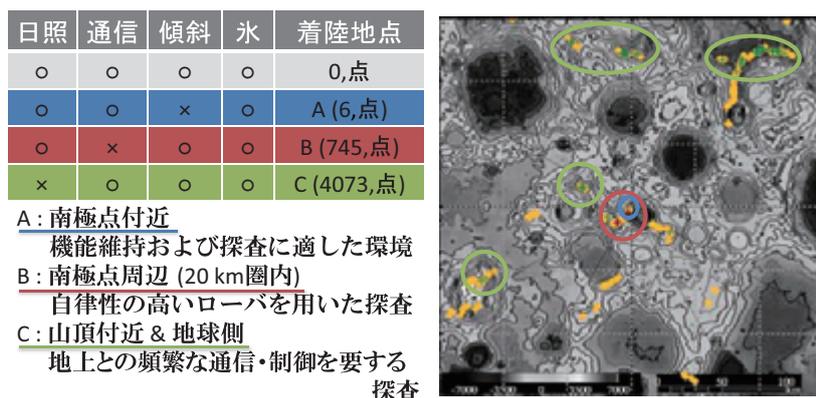


図5 多目的最適な着陸地点候補地点

これらの各地点について、目的関数ごとの傾向分類を行った。なお、以下で述べる目的関数の良好さは、目的関数値が0から0.5までの値を示したものとして定義している。

傾斜角以外の目的関数値が良好である地点は、図5における青で示した点である。発見された地点は南極点付近のみに限定的に存在し非常に数が少ないものの、日照や通信、さらに科学的な観点から有意義な氷の存在が示唆されるため、着陸出来れば探査機の機能維持および探査に非常に適した環境であるといえる。

次に通信以外の目的関数値が良好である地点は、図5における赤で示した点である。南極点を中心とする半径20km圏内に分布しているのは、南極点周辺に大きな山などの障害物があまり存在せず、一方でクレーターなど永久影の存在しやすい条件が揃っていたためであると考えられる。通信以外の目的関数が良好であるため、地上からの頻繁なコマンド送信を必要としない自律性の高いローバを用いた探査を行うようなミッションの着陸地点に適しているといえる。

最後に日照以外の目的関数値が良好である地点は、図5における緑で示した点である。小高い丘や山頂付近に点在しており、とりわけ図5の上部に多く分布しているのが見受けられる。これは図5の上部が地球側に面しており、通信のしやすい環境であるためだと考えられる。通信条件の良好さから、地上からの頻繁な制御や画像などのデータを大量に送る必要性のあるミッションなどに適した地点であるといえる。

5. まとめ

本研究では、日照、通信、傾斜角、氷の存在確率という4目的に対し多目的最適であるような月面探査機の着陸地点の選定を行い、その結果17413点の多目的最適な着陸候補地点が導き出された。さらにこれらについて目的関数ごとの傾向分類をすることにより、重視する目的ごとに選ぶべき着陸地点、またミッションの特徴についても有用な知見を得ることが出来た。

今後は制約条件により排除されている地点についても検討を加えたいと考えている。たとえば1つの制約条件をわずかに満たさないものの、それに関連しない目的関数については良好な値を示すような地点が存在する場合、これを見出すには制約条件を廃して全探索をするほかない。しかしながら9億点すべてについて各目的関数を評価していくのは計算負荷や時間の面からして現実的とは言えない。そのため探索に進化計算の手法を取り入れることで、計算コストを低減しなおかつ制約条件の境界に存在するような地点についても評価を行えるよう改良していきたいと考えている。

参考文献

- [1] C.Fonseca, P.Fleming, Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion and Generalization, Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, California, pp. 416-423, 1993.
- [2] 大山聖, 多目的設計探査と宇宙工学への利用, システム制御情報学会誌, 第55巻, 第9号, pp.374-381, 2011.
- [3] 立川智章, 野々村拓, 大山聖, 藤井孝蔵, 長田裕樹, 山本誠, ロケットの射点設計に向けた空力音響最適化問題の多目的設計探査, 日本機械学会 2013 年度年次大会, 岡山県岡山市, 2013 年 9 月.
- [4] 橋本樹明, 田中智, 星野健, 大嶽久志, 大槻真嗣, 月着陸・探査ミッション (SELENE-2) の現状について, 第 11 回宇宙科学シンポジウム, S3-05, 神奈川県相模原市, 2011 年 1 月.
- [5] S.Nolette, C.L.Lichtenberg, P.Spudis, R.Bonner, W.Ort, E.Malaret, M.Robinson, E.M.Shoemaker, The Clementine Bistatic Radar Experiment, Science, Vol.274, No.5292, pp.1495-1498, 1996.
- [6] Alan B. Binder, Lunar Prospector: Overview, Science, Vol.281, No.5382, pp.1475-1476, 1998.
- [7] D.A.Paige, M.A.Siegler, J.A.Zhang, P.O.Hayne, E.J.Foote, K.A.Bennett, A.R.Vasavada, B.T.Greenhagen, J.T.Schofield, D.J.McCleese, M.C.Foote, E.DeJong, B.G.Bills, W.Hartford, B.C.Murray, C.C.Allen, K.Snook, L.A.Soderblom, S.Calcutt, F.W.Taylor, N.E.Bowles, J.L.Bandfield, R.Elphic, R.Ghent, T.D.Glotch, M.B.Wyatt, P.G.Lucey, Diviner Lunar Radiometer Observations of Cold Traps in the Moon's South Polar Region, Science, Vol.330, pp.479-482, 2010.

JAXA キュレーションセンターにおけるリターンサンプルデータの管理システムの開発とその運用状況

○上棺 真之, 矢田 達, 唐牛 譲, 中藤 亜衣子, 熊谷 和也, 橋口 未菜子,
松本 徹, 岡田 達明, 安部 正真 (JAXA)

Development and operation status of data management system for Hayabusa-Returned samples in JAXA Extraterrestrial Sample Curation Center

Masayuki Uesugi, Toru Yada, Yuzuru Karouji, Aiko Nakato, Kazuya Kuamagai,
Minako Hashiguchi, Toru Matsumoto, Tatsuaki Okada, and Masanao Abe

Abstract

We developed a data management system for the analysis of Hayabusa-returned samples. The system consists of two major components, a work log management system and sample database system. Analyzed data of Hayabusa-returned samples were picked up from the work log management system, and converted and inputted into the sample database system. Through the system, we provided such sample information for the international announcement of opportunity of Hayabusa sample investigation. We will add the sample database of returned samples of future sample return missions, such as Hayabusa 2, into the system in the future work.

Keywords: extraterrestrial sample curation, Hayabusa-returned samples

概要

本論文では JAXA 地球外試料キュレーションセンターにおける、データの管理システムの開発と、現状の機能について説明する。このシステムは、キュレーションで実施される作業を管理する作業管理システムと、「はやぶさ」帰還試料のデータベースシステムの二つの要素で構成される。作業管理システムに登録された作業データのうち、「はやぶさ」帰還試料の分析データをシステムが抽出し、サンプルデータベースに追加する。2011 年の運用開始以降これまでに 3 回行われてきた「はやぶさ」帰還試料の国際公募研究に対して、試料を選定する情報を提供するなど、本システムは大きな成果を上げている。今後は「はやぶさ 2」帰還試料や、その他の帰還試料のサンプルデータベースを既存システムに追加する予定であり、さらに海外機関との連携も視野に入れた分析データ公開システムの構築を検討している。

1. はじめに

地球惑星科学分野、特に地球外物質を扱う惑星物質科学分野において、試料のデータベースシステムの開発は他の宇宙科学分野と比べて進んでいるとはいえない。この背景として(1) 試料を回収できる場所や手法が限られており、試料の絶対量が少ないため、データベース開発が盛んな材料工学や生化学分野と比べて分析例が多くないこと、(2) 多くの場合、地球外試料は化学組成・組織等にきわめて高い不均質性をもっているため、分析成果が研究成果につながりやすく、論文化せずに分析結果を他者に提供することが少ないこと、(3) 天然試料の分析は、たとえ同じ装置で分析したとしても、試料準備や分析フローなどによって得られる結果(解釈)が異なる場合があるため、統一された手法で

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

の分析結果や他の研究に直接応用できる分析結果は少ないこと、(4) 鉱物学・岩石学等の地球科学だけでなく、工学、生物学、化学など、数多くの分野にまたがる分析手法、装置を利用しており、分析の結果得られるデータのフォーマットも膨大な種類が入り乱れていること、(5) 以上の理由から、他者が分析したデータを多くの作業を介して再利用して研究を行うメリットが少ないこと、(6) フィールドワークを主とする地質学分野出身の研究者が多く、業界全体がデータベースシステムの開発経験そのものに乏しいこと、(7) 慢性的な予算不足・人手不足により、データ管理にまでコストが避けられないこと、等の理由が挙げられる。これらのうち(1), (2)は他の宇宙科学分野とだけでなく、地球物質分野とすらも大きく異なる、惑星物質科学分野特有の背景といえる。端的に言うと、データベースそのものが殆ど求められてこなかった。

しかし、この惑星物質科学分野でも、試料データベースシステムが必要となる場面がある。天然試料を収集し、その後保管・管理・配分し、科学成果を創出する立場にある、キュレーション(curation)と呼ばれる活動である。National Aeronautics and Space Administration, Johnson Space Center (NASA JSC) のキュレーション施設では、アポロの月サンプル、Genesis の太陽風サンプル、スターダストの彗星試料等、これまでの NASA の地球外サンプルリターン計画で得られた試料を管理しており、その他に成層圏フライトによって収集できる微隕石(interplanetary dust particles, IDP)や南極隕石などもプロジェクトとして収集している [1]。こういった試料に加え、JAXA から配分された「はやぶさ」試料や、探査に使われた人工衛星の表面物質などを同時に保管し、管理・配分している。隕石試料や IDP は保有試料のリストが Web 上の表や Excel シートで提供されている。その他のリターンサンプルは PDF やフォトギャラリーなどの形で画像データとともに、組成や組織等の詳細な情報が配分に対する参考資料として提供されている。

日本では国立極地研究所の南極隕石ラボラトリーが南極で回収した世界最大級の隕石コレクションを管理しており、これまでの所、国内唯一の地球外試料のキュレーション施設であった。試料は薄片の作成、化学組成分析などが実施され、これらのデータが PDF リストの形で定期的に Web 上に公開されている [2]。

このほかに、Natural History Museum of London や American Museum of Natural History 等でも地球外試料のキュレーションが行われている。このような施設では、回収された隕石試料の配分のために、リストを Web や書籍の形で公開している。また、キュレーション作業ではないが、国際隕石学会(The Meteoritical Society)では、世界中で得られた隕石の承認(名前の付与)を 1957 年から行っており、その過程で得られた情報を Meteoritical Bulletin [3] 及び Meteoritical Bulletin Database [4] という形で公開している。これらのデータベース・公開資料において多くの場合、重量、サイズ、発見場所や時期、隕石タイプ(普通コンドライト、炭素質コンドライトなど)、グループ(H, L, LL, EH, EL, CV, CM, CI 等)のほか、風化度などの情報が提供されている。

しかし、これらのキュレーション活動においても、試料データベースの開発には多くの課題が存在する。上で述べたとおり、地球外試料は単位試料量が少なく、試料を配分する前に破壊し、消耗し尽くすわけにはいかないため、収集だけでなく、保管・配分まで行うキュレーション活動における分析手法は限られている。また、高精度な分析手法になるほど、試料の前準備、及び分析そのものに多くの技術と時間をかける必要があるため、収集された試料すべてに高精度分析手法を適用するのはコストがかかりすぎる。さらに、キュレーションを行う施設のスタッフも研究者であるため、分析して得られたデータを論文として発表する前に一般に公開することによるジレンマも存在する。このように、実際に試料データベースシステムを必要とするキュレーション活動においても、データの取得、公開には多くのハードルが存在し、データベースシステムの開発は進んではいなかった。近年では惑星物質学の試料管理に応用可能なデータベースシステムの開発及びそれを利用したデポジトリの開発の活動報告がなされているが [5]、実際のデータの一般公開・試料配分に対する運用には至っていない。

このような背景の中、2010 年に Japan Aerospace exploration Agency (JAXA)は小惑星探査機「はやぶさ」によって世界初の小惑星表面からのレゴリス粒子のサンプルリターンに成功した。本論文では、JAXA 地球外試料キュレーションセンター(Extraterrestrial Sample Curation Center, 以降

ESCuC) における「はやぶさ」帰還試料の記載データの管理システム、及びサンプルデータベースシステムの開発について報告する。

2. ESCuC におけるデータ管理システム開発の経緯

「はやぶさ」帰還試料 (Hayabusa-returned samples)の分析は大きく三つのフェーズに分けられる。初期記載(initial description)と初期分析(preliminary examination)、そして詳細分析(detailed analysis)である。初期記載は ESCuC でキュレーションチームが実施する記載フローで、すべての「はやぶさ」帰還試料が対象である。この初期記載で「はやぶさ」帰還試料は組成に基づき、以下の4つのカテゴリに分類される [6]。小惑星イトカワの表面から採取されたレゴリス粒子 (Itokawa particles) はカテゴリ1と2、主に炭素から構成されるカテゴリ3、人工物から成ると思われるカテゴリ4である。初期分析は、このうちの「イトカワ粒子」、すなわちカテゴリ1と2、に対して国内の共同チームによって「はやぶさ」帰還後1年で実施された、より詳細な記載・分析作業である [7-13]。近年、炭素から構成されるカテゴリ3についても別途初期分析が実施されている [14-18]。最後に、詳細分析のフェーズがあり、これは国際公募研究(AO)や JAXA 主導のコンソーシアム研究などが当てはまる [6]。この詳細分析は現在の所、期間は制限されておらず、公募に対する申請に応じて、審査の上試料の配分が実施される。

初期記載の分析データはその後の初期分析、及び詳細分析の試料選択に使用するもので、「はやぶさ」帰還試料の全容把握のためにも重要な情報である。2010年の「はやぶさ」帰還当初、ESCuCにはこのデータの明確な管理システムが存在しておらず、得られたデータをプリントアウトしたものにペンでノートを書き込み、バイндаで綴じて管理していた。従って、初期分析にはこのバイндаによる初期記載情報が提供された。しかし、データの増加と共にデジタルデータの検索が困難になり、試料レポート作成にも膨大な時間を要した。さらに、データの取り違いや紛失など、トラブルも頻発しはじめた。それらの解消のため、2011年7月にデータの管理システムの開発が始まった。開発は「はやぶさ」帰還試料のキュレーション作業と並行して実施され、同8月初頭に運用を開始した。運用開始直後から11月までは稼働開始以前のデータの登録も並行して行われている。なお、第1回の国際AOは2012年の1月に発行されており、システム開発の開始からこのAOでの試料情報公開までに半年しか要していない。このように本システムは極めて短い開発期間・運用までの移行期間であったにもかかわらず、その後の国際AOおよびキュレーション作業に多大な貢献をしている。この短期間での開発と運用が可能だった理由として、すでに応用可能な基礎的なデータベースシステムが主著者により開発されていたこと [19-20]、また当時の記載データ量が極めて限られていたため、最小限の調整で適用可能であったこと、第三者を介さず、キュレーション作業員である主著者が直接システムを構築したため、要件の確認などの手間を短縮できたこと、後述の通り、それまでの作業との親和性を考慮した運用システムを導入したことなどがあげられる。

なお、システムの改修はその後も並行して続けられ、2015年現在も高速化やインターフェースの改良など、システムアップデートは続けられている。

3. システム

本システムは大きく二つのシステムから構成されている。ひとつは作業管理システムであり、これは ESCuC のクリーンルーム内の作業で得られるすべてのデータを管理する。ESCuC では「はやぶさ」帰還試料を扱う作業以外にも、設備メンテナンスなどの作業も実施されており、それらのデータなどもすべてネットワークを通じてネットワークストレージにアーカイブされる。試料の分析データを含むこれらの作業データは、Web上の統一されたシステムにより、テキストデータを付加され、集中管理される。もう一つはサンプルデータベースシステムであり、上記の作業管理システムに保存されたデータの中から「はやぶさ」帰還試料に関する記録をピックアップして、Web上のシステムを介してより詳細な試料情報を付与したのち、試料情報検索に特化して使用される。

データ保存後、作業者はクリーンルーム外のデータ管理用 WindowsPC で、Web ブラウザを介して保存したデータにアクセスする。クリーンルーム内は常に全身を覆うクリーンスーツを着用しており、さらにクリーンルームの外気をコントロールするフィルターファンユニットによる轟音が響いているため、作業には恒常的に極めて高い負荷がかかっている。このため、滞在時間は必要最小限とし、時間がかかるデータ入力作業はクリーンルーム外で行う方針になっている。この時間差によって発生する入力ミスなどは、後述の管理者によるチェックで対処している。サーバープログラムはネットワークストレージの階層構造を参照し、“作業内容”のフォルダ名を解析して作業内容をグループ分けし、表示する(図 1)。さらに、各“作業内容”フォルダの項目にはリンクが張られており、作業内容の詳細ページを表示することが出来る。作業内容詳細ページでは、作業者は作業情報を登録、閲覧することが出来る(図 2)。作業情報の一部として試料 ID を登録すると、図 1 の作業一覧の各カラムにその試料 ID が表示され、各試料に対する作業日時と作業内容が一覧で確認出来る。作業情報閲覧画面では、保存された分析データを閲覧しやすい形にフォーマットし直すプラグインシステムが採用されている。これにより、本来 Web での閲覧には適さない Word フォーマットなどで保存される各分析装置のレポートなどを、図 2 のように必要な情報だけ適切な形で表示することが可能になる。

サーバープログラムは、CGI を介して渡された各データに対する入力情報を、同様の階層構造を持ったサーバー上のデータベースに、テキストデータとして保存する。入力されたデータはいつでも入力画面を通して更新が可能であり、新しいファイルが追加された場合・削除された場合はユーザーが該当ディレクトリにアクセスした際に検知されるため、同様に入力が可能である。また、後述のサンプルデータベースへの変換の際、必ずファイルが存在しているかどうかの確認を行っているため、ファイルが存在しない場合はサンプルデータベースには登録されず、新規ファイルが追加された場合は新しい情報が追加される。本システムの問題点として、作業フォルダの変更が難しい点がある。作業フォルダの名前などを変更した場合、その内部にあるすべてのファイルのツリーが変更になるため、作業データをすべて再度入力する必要がある。現在は管理者がデータファイルを直接編集して対応しているが、将来的にはユーザーが修正可能なようにシステムを改良する必要がある。

3.2. サンプルデータベース

サンプルデータベースシステムは、作業管理システムのデータを検索し、新しい試料 ID を見つけると、この試料のレコードを自動で作成し、その作業記録データの階層情報をフィールドに追加する。この新しい試料 ID の付与は粒子が電子顕微鏡による記載を終え、保管先のスライドガラス上に移動した時点で行われ、それまでは粒子は仮 ID で管理される。この仮 ID はサンプルデータベースには反映されない。従って、粒子のスライドガラス移動後、電子顕微鏡による記載前の拾い出しの時点まで遡って、仮 ID を正式な試料 ID へ置き換える作業が発生する。管理者はこの ID の置き換え作業時に、日々の登録・記載情報に誤りが無いか、粒子の入れ違い等のチェックを実施する。実際にはこの時点でかなりの登録ミス・記載の混乱が発見され、訂正されている。このため、この仮 ID と正式な粒子 ID の置き換え作業は自動化も可能だが、現在は月 1 回のペースで手作業で、この新規レコードの追加を行っている。

サンプルデータベースには、この作業情報の他に、現在の保存場所、試料状態、移動履歴、サイズ、カテゴリ、鉍物組成、分析履歴、コメントなどのフィールドが存在する(図 3)。

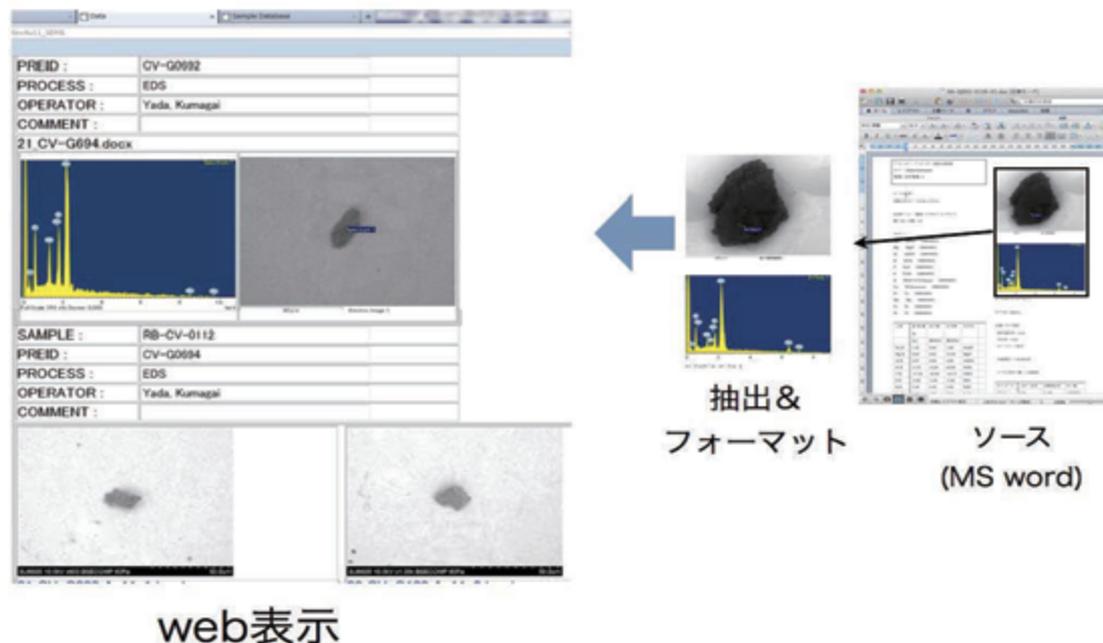


図 2 作業管理システム詳細画面。作業内容の登録・閲覧をおこなう。例では「はやぶさ」帰還試料の分析データを登録している。プラグインにより、MS Word のデータを抽出して成形し、Web 画面上に表示している

開発当初は新規の試料データ検索をユーザーがデータベースにアクセスした際に毎回行っていた。これは作業者が誰でも新規データ入力する事が可能で、作業負荷の分散を目的としたシステムである。しかし作業管理システムのデータ件数が増えるにつれ検索時間が増大し、その間はデータベースでの作業が出来なくなる上、複数の作業による入力のためのミス、混乱が増大した。これは第 1 回の国際 AO の試料配分時に大きな問題となった。このため、上記で説明したように、新しい試料レコードの作成作業を、単一の管理者が定期的にまとめて実施し、データ入力のチェックを行い、その作業直後にデータベースのアップデートを実施することで運用する方針に変更した。

サンプルデータベースでは試料レコードの各フィールドを利用しての絞り込み検索が可能だが、本システムは当初、保存されているすべてのデータに対して、ループ文で検索を行う、きわめて効率の悪い手法をとっていた。データ量の少なかった開発当初はこれで十分な速度であり、単一のフィールドへの入力での全文検索が可能などのメリットもあった。しかしその後、試料ピックアップの効率化と共に予想を超えてデータ量が一気に増大したため、検索に非常に時間を要するようになった。このため、すべての情報を一括保存した主データの他に、検索に使われる粒子情報のテキストデータ、各バイナリに付与されるメタデータ、保存されたデータのディレクトリツリーのデータなどに細分した中間キャッシュを作成するよう、システムを変更した。検索時には検索項目を指定し、個々のデータを含むキャッシュに対してのみ検索を行う事でパフォーマンスの向上を行うよう、検索エンジンの変更をおこなった。この結果、検索時間は最大時で 1 分近くかかっていたが、現在は検索項目数によらず 1 秒以下になっている。この中間キャッシュはサンプルデータベースへの新規情報の登録の際に主データと一緒に更新される。

データベースの検索で試料レコードを抽出した後は、作業管理システムで保存された分析データや、サンプルデータベースシステムから入力された試料情報などを閲覧出来る (図 4)。サンプルデータベースでの分析データの閲覧にも、作業管理システムで使用されている表示用のプラグインを利用することが出来る。

現在の所、ESCuC で初期記載に使用しているのは電界放出形走査電子顕微鏡 (Field emission-type scanning electron microscope, FE-SEM) 及びエネルギー分散型 X 線分析装置 (energy dispersion X-

国際 AO など、「はやぶさ」帰還試料のデータを公開する場合、上述の通り CSV,PDF 等のフォーマットでデータを打ち出した後、別の Web アプリケーションでそれを読み込む形で利用する。データベースサーバーとデータ公開サーバーを物理的に切り離しているため、情報の即時公開などには適さない。しかし、キュレーションのシステムのセキュリティを維持するには必要なことである。また、公開の前にデータの検証など最終確認を行えるメリットもある。実際に国際 AO へのデータ提供の際には、タイプミスや前後矛盾などを修正するためのチェックが行われている。

開発開始時にすでに「はやぶさ」帰還試料のキュレーション作業が始まっていたこともあり、本システムの開発において最も重要視されたのが、可能な限り早く、また円滑に運用を開始することであった。このために、本システムは、既存のデータ運用フローを可能な限り吸収する形で開発が進められている。本システム開発以前から、データはネットワークストレージ上に、名前に日付の入ったフォルダに保存されていた。フォルダには非常に大雑把なルールによる、作業内容が判別可能な名称が日付の後に付加されていた。本システムでのデータ保存は、既存のフォルダの名前付けと階層に関してより明確なルールを適用しただけであり、作業への運用開始時の負担を必要最小限にとどめている。

図 4 サンプルデータベース詳細画面。各試料の分析データ、ハンドリング履歴や、その詳細を閲覧できる。

また、システム開発以前は作業記録およびサンプルカタログはプリントアウトした物を共有していた。本システムでは運用開始後も同様の作業が可能なように、作業管理システムやサンプルカタログシステムの詳細画面は、Web ページをそのまま印刷すれば、レポートとして適切な形に再フォーマットされるよう設計している。

初期記載での FE-SEM/EDS による分析結果は分析装置のソフトウェアが作成するレポート機能を利用して、MS Word 形式のファイルフォーマットで保存されていた。本システムではプラグインシステムを採用した結果、作業者はそれまで通りのフローでデータを保存すれば、作業管理システム及びサンプルデータベースシステムを通して Web ブラウザでそのデータの閲覧が可能となる。

作業管理システムに入力するデータが多いほど、その後より多くの作業に応用できるが、一回の入力作業に割く負担が増大する。ほぼ毎日膨大な作業データが発生するキュレーション作業においては、入力データを必要最小限に絞り、まずは運用開始の早期化を優先した。そのため、運用開始当初は作

業管理システムの入力項目は試料名と作業内容の 2 項目のみで、まずは作業者が入力作業に慣れることからスタートした。その後項目を徐々に増やして、現在の 5 項目（試料 ID、仮試料 ID、作業名、作業内容、コメント）としている。また、サンプルデータベースでの試料の詳細情報の入力も管理者が適宜実施し、毎日の作業とは切り離されている。

以上のような取り組みの下、データベースシステム、ネットワークシステムあるいは PC システムについて必ずしも全員が精通しているわけではない ESCuC チームメンバーに対して、運用に伴う負担を可能な限りシステムが吸収する形で最小限にすることで、短期間の内に運用を軌道に乗せることが可能となった。

4. 今後の課題

現在本システムで取り扱っているのは「はやぶさ」帰還試料のみだが、今後 ESCuC では「はやぶさ 2」等の新たな帰還試料の受け入れを予定しており、これらに対応したデータベースが必要になる。作業管理システムは、帰還試料が増える毎にインターフェースを微調整することで、すべての帰還試料の作業に対して共有できる。このインターフェースの調整は、すでに現行のシステムで管理者が任意にカスタマイズすることが可能になっている。サンプルデータベースシステムは作業管理システムに記録された各帰還試料に対して個別のデータベースを独立して設置できるため、今後の帰還試料の増加に対しても、十分な柔軟性を持って対応出来る。

現在のシステムでは、サンプルの分析データはプラグインで加工後に PDF 化して提供する仕組みになっているが、この場合スペクトルデータをグラフ化後の画像フォーマットでしか提供できないなど、限界がある。今後は分析データをそのまま提供することを検討しており、JAXA の科学衛星運用・データ利用センター (C-SODA) と連携し、試料データアーカイブを公開する予定である。この場合、ESCuC で作成する PDF を利用したカタログは、分析結果の一部をプラグインで加工した試料の概要情報とし、詳細データ（生データ）を同時に提供する形となる。具体的な公開方法は現在検討中である。

さらに、今後の帰還試料分析に向けて、新しい試みも検討している。現在のサンプルデータベースはカタログとしての機能に限定しているが、公開予定の試料データアーカイブはすべてのデータを利用可能な形で公開することから、これらを統計的に扱い、新たな科学成果を創出できる研究を目的としたアプリケーションシステムの開発を検討している。ESCuC では X 線回折装置や X 線 CT 装置など非汚染・非破壊の分析手法を開発し、初期記載にこれらを組み込むことで、試料データを拡充する予定である。これらは、研究者が試料を選ぶ際、より適切な試料を選定することに貢献できる。一方で、非汚染・非破壊で得られる初期記載データは、精度としては破壊分析に劣るため、それだけで科学的成果を創出するのは難しい。しかし一連のデータを統計的に扱うことが出来れば、試料全体を新しい視点で俯瞰することが可能になる。これらの分析装置から得られるデータの統計的な取り扱いには手法のさらなる研究開発が必要だが、例えば X 線 CT の場合、隕石試料内部の包有物のサイズ分布や、化学組成と相関する X 線吸収係数の分布などを利用して複数の隕石試料を系統的に議論する試みはすでに行われている [23-24]。

今後海外でもサンプルリターン計画が予定されており、アメリカでは 2023 年に帰還予定の OSIRIS-REx が 2016 年に打ち上げ予定である。ESCuC は、「はやぶさ 2」と OSIRIS-REx の試料受け入れ準備を NASA JSC と連携して実施することを計画している。汚染物質の情報など、お互いの予備分析の情報を公開データアーカイブ等を通して円滑に共有できるシステムを開発することができれば、両者の試料受け入れ準備に大いに貢献できる。例として、海洋研究開発機構 (JAMSTEC) では、海底のボーリングコアサンプルのデータを Integrated Ocean Drilling Program (IODP) に所属する複数の大陸の研究機関で共有するシステムがすでに稼働している [25]。これらを参考にし、現在 C-SODA との共同で開発中のデータアーカイブシステムを中心として、今後はできる限り多くの JAXA

内部機関、及び外部機関との連携を行い、複数国にまたがる研究機関の連携を促進する、惑星試料データベースシステムの構築を目指したい。

5. まとめ

本論文では ESCuC での、「はやぶさ」帰還試料を対象にした作業管理システムおよびサンプルデータベースシステムを構築する過程、およびその内容について紹介した。惑星試料のデータベースシステムは、これまでの研究者個人の研究ベースでは労力に見合う成果が見込めないためあまり必要とされず、キュレーション活動においても制約が多いため、積極的な開発は行われてこなかった。しかし、今後サンプルリターン計画が活発化し、惑星物質の分析データが広く共有される機会が増えれば、このようなデータベースシステムの開発を促し、個人の研究にも簡便に適用する事が可能になり、惑星物質学に新しい視点をもたらすことが期待される。

謝辞

本論文の作成においては、匿名の査読者にきわめて有益なアドバイスを多くいただき、著者の作業の大きな助けになった。この場を借りて感謝の意を表す。また、著者らはこの論文を作成する機会を与えてくださった海老沢研氏、山本幸生氏に感謝する。

References :

- 1) NASA astromaterials Acquisition and Curation, <http://curator.jsc.nasa.gov>, Johnson Space Center, National Aeronautics and Space Administration (NASA JSC), USA
- 2) Antarctic Meteorite research Center, National Institute of Polar Research, <http://yamato.nipr.ac.jp/en/>, National Institute of Polar Research (NIPR), Japan
- 3) Meteoritical Bulletin Archive, http://meteoriticalsociety.org/?page_id=57, The Meteoritical Society
- 4) Meteoritical Bulletin Database, <http://www.lpi.usra.edu/meteor/metbull.php>, The Meteoritical Society
- 5) 北川宙, 谷内勇介, 国広卓也, 中村栄三: 地球惑星試料デポジトリ DREAM の構想と現状, 宇宙科学情報解析論文誌 第3号 (2013) 71-77, <https://repository.exst.jaxa.jp/dspace/handle/ais/16896>
- 6) Yada T., Fujimura A., Abe M., Nakamura T., Noguchi T., Okazaki R., Nagao K., Ishibashi Y., Shirai K., Zolensky M. E., Sandford S., Okada T., Uesugi M., Karouji Y., Ogawa M., Yakame S., Ueno M., Mukai T., Yoshikawa M., and Kawaguchi J. Hayabusa-returned sample curation in the Planetary Material Sample Curation Facility of JAXA, Meteorit. Planet. Sci. 49, 135-153, (2014)
- 7) Nakamura T., Noguchi T., Tanaka M., Zolensky M. E., Kimura M., Tsuchiyama A., Nakato A., Ogami T., Ishida H., Uesugi M., Yada T., Shirai K., Fujimura A., Okazaki R., Sandford S. A., Ishibashi Y., Abe M., Okada T., Ueno M., Mukai T., Yoshikawa M., and Kawaguchi J., Itokawa dust particles: a direct link Between S-Type Asteroids and Ordinary Chondrites, Science, 333, 1113-1116 (2011)
- 8) Yurimoto H., Abe K., Abe M., Ebihara M., Fujimura A., Hashiguchi M., Hashizume K., Ireland T. R., Itoh S., Katayama J., Kato C., Kawaguchi J., Kawasaki N., Kitajima F., Kobayashi S., Meike T., Mukai T., Nagao K., Nakamura T., Naraoka H., Noguchi T., Okazaki R., Park C., Sakamoto N., Seto Y., Takei M., Tsuchiyama A., Uesugi M., Wakaki S., Yada T., Yamamoto K., Yoshikawa M., and Zolensky M. E., Oxygen isotopic compositions of asteroidal materials returned from Itokawa by the Hayabusa mission, Science, , 333, 1116-1119 (2011)

- 9) Ebihara M., Sekimoto S., Shirai N., Hamajima Y., Yamamoto M., Kumagai K., Oura Y., Ireland T. R., Kitajima F., Nagao K., Nakamura T., Naraoka H., Noguchi T., Okazaki R., Tsuchiyama A., Uesugi M., Yurimoto H., Zolensky M. E., Abe M., Fujimura A., Mukai T., and Yada T., Neutron activation analysis of a particle returned from asteroid Itokawa, *M. Science*, , 333, 1119-1121 (2011)
- 10) Noguchi T., Nakamura T., Kimura M., Zolensky M. E., Tanaka M., Hashimoto T., Konno M., Nakato A., Ogami T., Fujimura A., Abe M., Yada T., Mukai T., Ueno M., Okada T., Shirai K., Ishibashi Y., and Okazaki R.. Incipient Space Weathering Observed on the Surface of Itokawa Dust Particles. *Science* 333: 1121-1125. (2011)
- 11) Tsuchiyama A., Uesugi M., Matsushima T., Michikami T., Kadono T., Nakamura T., Uesugi K., Nakano T., Sandford S. A., Noguchi R., Matsumoto T., Matsuno J., Nagano T., Imai Y., Takeuchi A., Suzuki A., Ogami T. , Katagiri J., Ebihara M., Ireland T. R., Kitajima F., Nagao K., Naraoka H., Noguchi T., Okazaki R., Yurimoto H., Zolensky M. E., Mukai T., Abe M., Yada T., Fujimura A., Yoshikawa M. and Kawaguchi J., Three-dimensional structure of Hayabusa samples: Origin and evolution of Itokawa regolith, *Science*. , 333, 1125-1128, (2011)
- 12) Nagao K., Okazaki R., Nakamura T., Miura Y. N., Takahito O., Bajo K., Matsuda S., Ebihara M., Ireland T. R., Kitajima F., Naraoka H., Noguchi T., Tsuchiyama A., Yurimoto H., Zolensky M. E., Uesugi M., Shirai K., Abe M., Yada T., Ishibashi Y., Fujimura A., Mukai T., Ueno M., Okada T., Yoshikawa M., and Kawaguchi J., Irradiation history of Itokawa regolith material deduced from noble gases in the Hayabusa samples, *Science*, , 333, 1128-1131, (2011)
- 13) Nakamura E., Makishima A., Moriguti T., Kobayashi K., Tanaka R., Kunihiro T., Tsujimori T., Sakaguchi C., Kitagawa H., Ota T., Yachi Y., Yada T., Abe M., Fujimura A., Ueno M., Mukai T., Yoshikawa M., and Kawaguchi J. 2012. Space environment of an asteroid preserved on micrograins returned by the Hayabusa spacecraft. *Proceedings of the National Academy of Sciences of the United States of America* 109: E624-E629.
- 14) Ito M., Uesugi M., Naraoka H., Yabuta H., Kitajima F., Mita H., Takano Y., Karouji Y., Yada T., Ishibashi Y., Okada T., Abe M. (2014) H, C and N isotopic compositions of HAYABUSA Category 3 organic samples. *Earth, Planet. Space.* 66:91,
- 15) Uesugi M., Naraoka M., Ito M., Yabuta H., Kitajima F., Takano Y., Mita H., Ohnishi I., Kebukawa Y., Yada T., Karouji Y., Ishibashi Y., Okada T., Abe M. Sequential analysis of carbonaceous materials in Hayabusa-returned samples for the determination of their origin. *Earth, Planets. Space.* 66:102, (2014)
- 16) Yabuta H., Uesugi M., Naraoka H., Ito M., Kilcoyne A. L. D., Sandford S. E., Ohigashi T., Kitajima F., Mita H., Takano Y., Karouji Y., Yada T., Ishibashi Y., Okada T., Abe M. Molecular compositions of Hayabusa Category 3 carbonaceous particles. *Earth, Planet. Space.*, 66:156 (2014)
- 17) Kitajima F., Kotsugi M., Ohkochi T., Naraoka H., Ishibashi Y., Uesugi M., Karouji Y., Abe M., Fujimura A., Yada T., Okazaki R., Nakamura T., Noguchi T., Nagao K., Tsuchiyama A., Yurimoto H., Ebihara M., Ito M., Yabuta H., Mita H., Takano Y., Mukai T., Sandford S. A., Okada T., Shirai K., Ueno M., Yoshikawa M., Kawaguchi J. A micro-Raman and infrared spectroscopic approach to the several stony and organic (category 3) particles recovered by the Hayabusa mission, *Earth, Planet. Space.* 67:20 (2015)
- 18) Naraoka H., Aoki D., Fukushima K., Uesugi M., Ito M., Kitajima F., Mita H., Yabuta H., Takano Y., Yada T., Ishibashi Y., Okada T., Abe M. ToF-SIMS analysis of carbonaceous particles in the sample capsule of the Hayabusa mission. *Earth, Planet. Space.* 67:67(2015)

- 19) 上梶真之, 上杉健太郎, X 線 CT の 3 次元データのためのデータベース開発, 日本地球惑星科学連合大会予稿集, MGI015-06 (2010)
- 20) Uesugi H. and Uesugi M., Development of the Database for Images of the Text on the Stone Monuments. In Culture and Computing (Culture Computing), 2013 International Conference on (pp. 149-150). IEEE. DOI: 10.1109/CultureComputing. 2013.40 (2013)
- 21) International Announcement of Opportunity for Hayabusa sample investigation, <http://hayabusao.isas.jaxa.jp>
- 22) Uesugi M., Yada T., Okada T., Karouji Y., Nakato A., Kumagai, K., Abe M. Hayabusa-returned sample catalogue 2014, JAXA Special Publication, JAXA-SP-14-007E, 1-770, <https://repository.exst.jaxa.jp/dspace/handle/a-is/236431>, (2015)
- 23) Uesugi M., Uesugi K., Takeuchi A., Suzuki Y., Hoshino M., and Tsuchiyama A., Three-dimensional observation of carbonaceous chondrites by Synchrotron radiation X-ray CT - quantitative analysis and developments for the future sample return missions -, *Geochimica et Cosmochimica Acta*, , 116, 17-32, (2013)
- 24) Uesugi M., Uesugi K. and Oka M., Non-destructive observation of meteorite chips using quantitative analysis of optimized X-ray micro-Computed Tomography, *Earth and Planetary Science Letters*, , 299, 359-367, (2010)
- 25) Core sample summary and Core sample availability, <http://www.kochi-core.jp/cs/> and <http://www.kochi-core.jp/sample-availability/>, Kochi Core Center, Japan Agency for Marine-Earth Science and Technology (JAMSTEC), Japan

全天 X 線監視装置「MAXI」のデータアーカイブの開発

中川 友進^{*1}, 海老沢 研^{*1}, 中平 聡志^{*1}, 上野 史郎^{*1}, 富田 洋^{*1}, 木村 公^{*1}
杉崎 睦^{*2}, 芹野 素子^{*2}, 三原 建弘^{*2}, 根来 均^{*3}

Developments of the MAXI Data Archive

Yujin E. Nakagawa^{*1}, Ken Ebisawa^{*1}, Satoshi Nakahira^{*1}, Shiro Ueno^{*1}
Hiroshi Tomida^{*1}, Masashi Kimura^{*1}, Mutsumi Sugizaki^{*2}, Motoko Serino^{*2}
Tatehiro Mihara^{*2}, Hitoshi Negoro^{*3}

Abstract

The Monitor of All-sky X-ray Image (MAXI) was launched on July 2009. It is installed on the exposed facility of the Japanese experiment module "Kibo" on-board the international space station (ISS). The main purpose of the MAXI is to discover and promptly report X-ray transients as well as steady monitoring of known X-ray sources in the whole sky. The MAXI operation is approved at least until the end of March 2018. We are working on public data release of the MAXI, in order to enhance use of the MAXI data by scientists all over the world. On-demand data analysis system using a web-interface has been in service since November 2011. Now we are working toward the permanent public MAXI data archive together with the MAXI specific software for scientific analysis.

Keywords: MAXI, Archive, Pipeline Processing

概要

MAXI は、全天における X 線突発変動天体の発見・速報、および既知の X 線天体の定常的な監視を目的とする全天 X 線監視装置である。2009 年 7 月に打ち上げられ、国際宇宙ステーションの「きぼう」日本実験棟の船外実験プラットフォームに搭載されている。2018 年 3 月まで運用が認められている。MAXI の観測データを全世界の科学者が容易に利用できることを目的として、データ公開を進めており、2011 年 11 月に開始した Web インタフェースを用いたオンデマンドデータの公開に続き、データを恒久的に後世に残すためのデータアーカイブの開発を進めている。データアーカイブの開発の現状、および今後の計画を報告する。

1 はじめに

全天 X 線監視装置「Monitor of All-sky X-ray Image (MAXI)」¹⁾ は、宇宙物理学の研究を行うために、宇宙の全ての方位を X 線で監視する装置である。MAXI はスペースシャトルにより 2009 年 7 月に打ち上げられ、国際宇宙ステーション (International Space Station; ISS) の「きぼう」日本実験棟の船外実験プラットフォームに搭載された。2009 年 8 月には初期運用を開始し、2009 年 10 月から 2012 年 10 月まで、当初の計画通りの 3 年間の定常運用を行なった。定常運用が終了するにあたって行われた延長審査において、運用延長が認められ、2015 年 3 月まで後期運用を行なった。現在は、さらなる運用延長が決定し、少なくとも 2018 年 3 月まで運用が認められている。

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

^{*1} 宇宙航空研究開発機構 (Japan Aerospace Exploration Agency)

^{*2} 理化学研究所 (Institute of Physical and Chemical Research)

^{*3} 日本大学 (Nihon University)

表 1: GSC と SSC の主な仕様

	GSC	SSC
視野	160 度 × 1.5 度 (2 方向)	90 度 × 1.5 度 (2 方向)
エネルギー帯域	2–30 keV	0.5–12 keV
エネルギー分解能	18% at 5.9 keV	<150 eV at 5.9 keV
位置決定精度	0.1 度	0.1 度

MAXI の主な科学目的は、ブラックホールなどの X 線天体の特性を明らかにすることである。この目的を達成するため、MAXI は新たな X 線天体を発見し、即座に研究者にその位置や明るさを速報するように設計されている。さらに、全天の既知の X 線天体を継続的に監視することができる。

新しい天体の探査、および既知の天体の活動状態の監視を行うために、日本大学に「リアルタイム突発天体探査システム」を設置している。図 1 に示すように、リアルタイム突発天体探査システムは、MAXIDB(TKSC/JAXA) からデータを取得する。

MAXI は、Gas Slit Camera (GSC)²⁾ と、Solid State Camera (SSC)³⁾ の 2 種類の X 線検出器で構成される。GSC は、1 個の Xe (99%) と CO₂ (1%) が充填された一次元位置感応型ガス比例計数管を "1 カメラ" と定義しており、12 カメラで構成される。一方、SSC は 16 個の X 線 CCD チップを "1 カメラ" と定義しており、2 カメラで構成される。したがって、32 個の X 線 CCD チップで構成される。X 線 CCD チップは 1 辺が 25cm (1024 ピクセル) の正方形の形状をしている。GSC と SSC の主な仕様を表 1 にまとめる。表 1 に記載されているように、GSC も SSC も扇型の細長い視野を持っている。そのため、ISS が地球を周回するのに合わせて、時々刻々と GSC と SSC の視野が変化していき、全天を掃くようにして、観測を行う。

MAXI は 2015 年 12 月時点において、6 年を超える長期間の観測を行っており、大量の観測データがあるため、MAXI チームだけでは全ての観測データを詳細に解析できていない。MAXI チームによる日々の運用により、新しい天体の発見や、既知の天体の増光など、大部分の天体現象は科学コミュニティへ情報を発信できているが、暗い天体現象などは見落としている可能性がある。そのため、MAXI による科学的な成果を最大化するためには、MAXI の観測データやソフトウェアを全世界のエンドユーザー（科学者）へ向けて公開することが重要である。これによって、科学コミュニティの発展を期待できるだけでなく、既に出版された科学的な成果の再検証を可能にする。そのために、JAXA を中心として、RIKEN (理化学研究所)、大学の研究者からなる MAXI データアーカイブチームにより、観測データの恒久的な保管、および全世界のエンドユーザーが容易に利用できる解析システムの開発が進められている。本論文では、MAXI のデータアーカイブの基本設計、現在の状況や今後の計画を解説すると共に、MAXI アーカイブ開発から得られた教訓や、将来ミッションのアーカイブ開発に有用と思われる指針等についても述べる。

2 データアーカイブの基本設計

2.1 データアーカイブの要求

MAXI のデータアーカイブは、JAXA 宇宙科学研究所の科学衛星運用・データ利用ユニット (C-SODA) で運用されている Data Archives and Transmission System (DARTS) に保管され、全世界のエンドユーザーへ公開される。エンドユーザーの利便性、およびバックアップのため、データアーカイブのコピーを NASA の Goddard Space Flight Center (GSFC) が運営している High Energy Astrophysics Science Archive Research Center (HEASARC) に保管し、全世界のエンドユーザーへ公開する。データアーカイブの保管に際しては、高エネルギー宇宙物理学で標準的に用いられている Flexible Image Transport System (FITS) フォーマットを用いる。MAXI の運用終了後においても、データアーカイブは半永久的に C-SODA/JAXA と HEASARC/NASA において保管・公開され

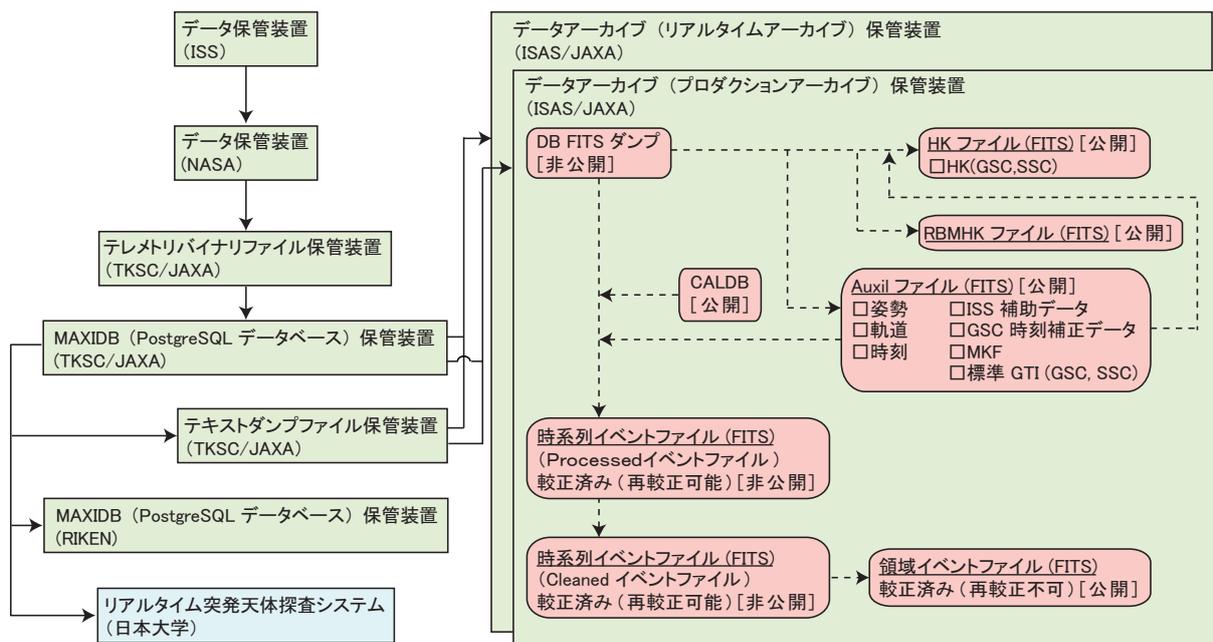


図 1: MAXI のデータが、ISS で取得されてから、データアーカイブへ保管されるまでのフローチャート、およびデータアーカイブの作成の流れを示す。図中の四角（薄緑）はデータ保管装置の名称、四角（薄青）は関連するシステムの名称、角丸四角（薄赤）はデータの名称を表す。四角（薄緑）や四角（薄青）を繋ぐ実線は、データ保管装置やシステムの間におけるデータの流れを表す。一方、角丸四角（薄赤）を繋ぐ破線は、どのような流れでデータが作成されるかを表す。リアルタイムアーカイブとプロダクションアーカイブの内容はほぼ同じであるが、更新頻度と完全性が異なる。

る。エンドユーザーは、HEASARC によってメンテナンスされる、HEASARC Software (HEASoft) と Calibration Database (CALDB)^{*4} を用いてデータ解析を行う。

データアーカイブで公開するデータには、公開の時点において最新の較正を適用する。また、再較正に必要なデータは公開しないため、エンドユーザーはデータを再較正することはできない。新しい較正が使用可能となった場合、全てのデータに再較正を行ない、新しいバージョンのデータアーカイブを公開する。

エンドユーザーの利便性を高めるために、DARTS と HEASARC は、ftp や http のプロトコルを用いて、任意の天体座標や観測期間のデータをダウンロードできるサービスを提供する。より高度なユーザーインターフェースを用いたデータの公開は、DARTS の JAXA Universe Data Oriented 2 (JUDO2^{*5}) から行う予定である。さらに、Web から科学プロダクツ(イメージ, エネルギースペクトル, 光度曲線, エネルギー応答関数) を得られるオンデマンド機能を、Universe via DARTS ON-line 2 (UDON2) に追加する。現在は、オンデマンド機能を RIKEN で公開しており、これを UDON2 へ取り込み、JUDO2 から呼び出せるようにする。

MAXI の低次データは、PostgreSQL を用いたデータベースである“MAXIDB”に保管している。MAXIDB を恒久的に保管することは、PostgreSQL を実行する環境の維持が難しいため、PostgreSQL のテーブルをそのまま FITS フォーマットへ変換した“DB FITS ダンプ”を作成し、源泉データとして保管する。FITS は仕様が決まっているので、半恒久的にデータを利用できる。MAXI は継続的に 1 周 (約 100 分) でほぼ全天を観測しており、そのためイベントデータは時系列に並んでいる。しか

^{*4} これらは MAXI チームが開発したものであるが、JAXA には全世界のユーザを対象に長期にわたってデータ解析環境をサポートする体制が整っていないため、HEASARC の協力を仰ぐことにした。

^{*5} <http://darts.isas.jaxa.jp/astro/judo2>

し多くの場合、エンドユーザーは、全天ではなく、より小さな領域のみのイベントデータを必要とする。そこで、Hierarchical Equal Area isoLatitude Pixelation of a sphere (HEALPix)⁴⁾ の手法を用いて、全天のイベントデータを 768 個の領域に分ける (領域イベントファイル)。領域イベントファイル、および解析に必要な姿勢や軌道の情報が書かれた Auxil ファイルは、毎日作成し、公開する。パイプライン処理を自動的かつ定期的に実行して、MAXIDB から領域イベントファイルや Auxil ファイルなどを作成する。

MAXI の情報は、様々な形態 (Wiki, メール, 口伝など) で残っており、それらを整理し、“アーカイブ基本設計書”として文書化し、後世の研究者やエンジニアが見たときに、MAXI のデータアーカイブの設計を把握できるように整備した。

MAXI の観測方法は特殊であり、後期運用 (2012 年 11 月～, §1 を参照) を開始してからデータアーカイブを開発・公開することが決まった。それまで用いられていたソフトウェアは、MAXI チームのみの使用を前提にしていたため、より簡便に利用できるソフトウェアの整備が必要であった。そこで、ソフトウェアの仕様や実行手順を整理して、容易に扱えるように改善した。さらに、エンドユーザーが円滑に科学解析を行えるように、科学解析の手順を再設計すると共に、自動的に GSC と SSC の科学プロダクツを作成するスクリプト (mxproduct) を開発した。データアーカイブの公開の際には、MAXI データアーカイブチームから、分かり易いマニュアルを提供する。

2.2 データアーカイブの構成

MAXI のデータは、10 秒程度の遅れで MAXIDB へ登録される “Real Time Data” (以後、REAL データ)、数十分から 6 時間程度の遅れで MAXIDB へ登録される “Communication Outage Recorder Data” (以後、COR データ)、数日から 1 ヶ月程度の遅れで MAXIDB へ登録される “欠損データ” の 3 種類に大別される。REAL データは全体の約 70% であり、残りの約 30% が COR データと欠損データである。COR データは原則として自動的に補完される一方、欠損データは手動で補完する必要があり、NASA のサーバーからデータを再取得することは容易ではない。また、図 1 に示す NASA より上流における欠損は補完が出来ない。データアーカイブを構築するにあたり、これらのデータの遅延を考慮する必要がある。そこで、MAXI のデータアーカイブは、データの完全性を保障しない “リアルタイムアーカイブ”、および取得できる全てのデータを使用した “プロダクションアーカイブ” の 2 種類に分けている。図 1 のデータアーカイブは、リアルタイムアーカイブとプロダクションアーカイブで構成される。どちらのデータアーカイブもファイル名やフォーマットは同じである。つまりエンドユーザーは、最新のデータはリアルタイムアーカイブから、完全性が保障されるデータはプロダクションアーカイブから、取得することになる。

リアルタイムアーカイブ

エンドユーザーが出来る限り最新のデータアーカイブを使用できるようにするため、リアルタイムアーカイブは 3 段階の更新頻度のデータで構成される。この 3 段階は、5 分毎に更新される “リアルタイム更新”、1 日毎に更新される “デイリー更新”、手動で更新する “手動更新” であり、更新頻度が遅くなるほどデータの完全性が高い。エンドユーザーはどの段階の更新頻度のデータなのかを意識すること無く科学解析を実行できるが、領域イベントファイルの FITS ヘッダーを見ることで、どの段階の更新頻度のデータなのか知ることができる。

プロダクションアーカイブ

プロダクションアーカイブは欠損データの補完が完了した時点においてのみ作成する。更新頻度は半年に 1 回程度である。MAXI データアーカイブチームによって手動で作成され、その際に領域イベントファイルの FITS ヘッダーにプロダクションアーカイブであることを記載する。

2.3 データの流れ

図1に示すフローチャートのうち、四角（薄緑）を繋ぐ実線で示している、MAXIのデータがISSで取得されてから、DARTSのデータアーカイブへ保管されるまでの流れを解説する。

データ保管装置 (ISS) → データ保管装置 (NASA) → テレメトリバイナリファイル保管装置 (TKSC/JAXA)

ISSで取得したテレメトリデータは、ISSからNASAの地上局へ1553bもしくはイーサネットのインターフェースを用いて送信される。これらのテレメトリデータは、インターネットを用いてNASAからJAXAのTsukuba Space Center (TKSC)に伝送され、テレメトリバイナリファイルとして保管される。

テレメトリバイナリファイル保管装置 (TKSC/JAXA) → MAXIDB(PostgreSQL データベース) 保管装置 (TKSC/JAXA) → MAXIDB(PostgreSQL データベース) 保管装置 (RIKEN)

MAXIDBはテレメトリバイナリファイルをデータの種類ごとにテーブルに分けて登録したPostgreSQLを用いたデータベースである。MAXIDBはTKSCとRIKENに配置され、それぞれ、MAXIDB(TKSC/JAXA)、MAXIDB(RIKEN)とよばれる。MAXIDB(RIKEN)は、MAXIDB(TKSC/JAXA)のコピーである。

MAXIDB(PostgreSQL データベース) 保管装置 (TKSC/JAXA) → テキストダンプファイル保管装置 (TKSC/JAXA)

テキストダンプファイルは、MAXIDB(TKSC/JAXA)のテーブルをそのままテキストにダンプしたものである。2.2節で解説する欠損データの補完が完了した期間についてのみテキストダンプファイルが生成される。

テキストダンプファイル保管装置 (TKSC/JAXA) → データアーカイブ保管装置

MAXIDB(TKSC/JAXA)、およびテキストダンプファイルから、データアーカイブが作成される。

2.4 データアーカイブの作成の流れ

MAXIDB(TKSC/JAXA)・テキストダンプファイル → DB FITS ダンプ

DB FITS ダンプは、MAXIDB(TKSC/JAXA)もしくはテキストダンプファイルを、テーブルごとにFITSフォーマットへ変換したものである。FITSフォーマットに最低限必要なヘッダー情報などを追記して、情報の損失やフォーマットの変更が無いように作成される。2.2節で解説したように、リアルタイム性を重視する場合はMAXIDB(TKSC/JAXA)から、欠損データ補完の完全性を重視する場合はテキストダンプファイルから、DB FITS ダンプが作成される。

DB FITS ダンプ → 時系列イベントファイル (Processed イベントファイル)

時系列イベントファイル (Processed イベントファイル) は、DB FITS ダンプのイベントファイルについて、検出器の較正データ (CALDB) と補助データ (Auxil ファイルの姿勢と時刻) を使って、生の値を工学値へ変換して、時刻、座標、エネルギーの情報を付け加えたものである。

時系列イベントファイル (Processed イベントファイル) → 時系列イベントファイル (Cleaned イベントファイル)

時系列イベントファイル (Cleaned イベントファイル) は、時系列イベントファイル (Processed イベントファイル) に対して、検出器の視野の端のデータを削除するなどのスクリーニングを行ったものである。

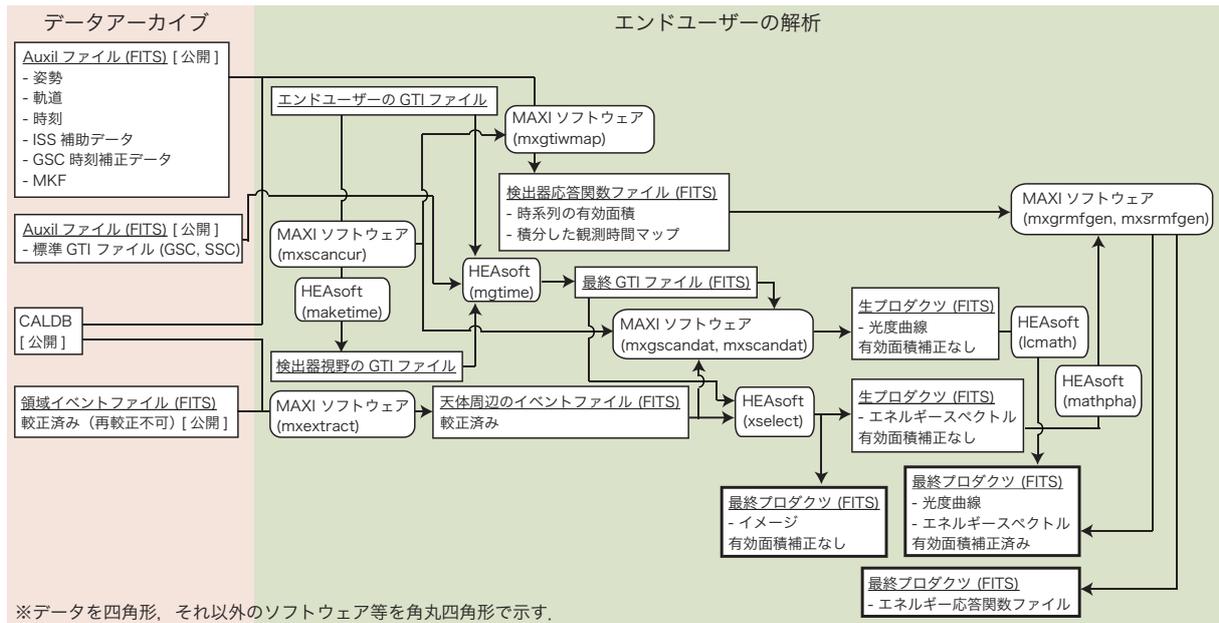


図 2: 左側（データアーカイブ）にエンドユーザーがデータアーカイブから取得するデータ，右側（エンドユーザーの解析）に mxproduct の内部で実行されるプロセスを示す。

時系列イベントファイル（Cleaned イベントファイル）→ 領域イベントファイル

領域イベントファイルは、全天を 768 個に分割して、時系列イベントファイル（Cleaned イベントファイル）を領域ごとに分けたものである。

CALDB

CALDB は地上較正や機上較正のデータを元に作成される。

DB FITS ダンプ → Auxil ファイル

DB FITS ダンプから、必要な情報を抜粋して、Auxil ファイルが作成される。Auxil ファイルは、MAXI の姿勢データが書かれた“姿勢”，ISS の軌道データが書かれた“軌道”，Data Processor の時刻補正が書かれた“時刻”，“ISS 補助データ”，“GSC の時刻補正データ”，時刻ごとの ISS の緯度・経度・高度，地球磁場の Cutoff-Rigidity，昼夜の情報が書かれた“MKF”，GSC と SSC の標準的な Good Time Interval (GTI) が書かれた標準 GTI で構成される。

DB FITS ダンプ → RBMHK ファイル

RBMHK ファイルは、DB FITS ダンプから、Radiation Belt Monitor のデータのみを抜き出したものである。

DB FITS ダンプ → HK ファイル（GSC, SSC）

HK ファイル（GSC, SSC）は、DB FITS ダンプから検出器（GSC と SSC）の House Keeping Data を抜粋して、Auxil ファイルの情報と合わせて、生成される。

2.5 エンドユーザーによるデータ解析

MAXI の観測方法は特殊であるため、解析の手順が複雑である。そこで、MAXI チームによる解析と同質の科学プロダクト（イメージ、エネルギースペクトル、光度曲線、エネルギー応答関数）を誰でも得られる解析の枠組みを MAXI アーカイブチームから提供する。具体的には、自動的に GSC と

表 2: mxproduct で実行される MAXI ソフトウェアの動作内容

ソフトウェア名	動作内容
mxextract	領域イベントファイルから、解析に必要な領域のデータを抜き出す。
mxscancur	エンドユーザーの GTI ファイルの時刻範囲における、検出器への天体の入射角度を計算する。
mxgtiwap	Auxil ファイルと mxscancur の出力ファイルを用いて、時系列の有効面積、および積分した観測時間マップが書かれた、検出器応答関数ファイルを作成する。
mxgscandat, mxsscandat	mxscancur の出力ファイル, 最終 GTI ファイル, 天体周辺のイベントファイルから、有効面積を補正していない光度曲線を作成する。
mxgrmfgen, mxsrmfgen	Auxil ファイルを用いて、エネルギー応答関数を作成する。mxgrmfgen が GSC 用, mxsrmfgen が SSC 用である。

表 3: mxproduct で実行される HEAsoft の動作内容

ソフトウェア名	動作内容
maketime	mxscancur の出力ファイルを使用して、天体が検出器の視野に入っている時間帯についての GTI (検出器視野の GTI ファイル) を出力する。
mgtime	エンドユーザーの GTI ファイル, 標準 GTI ファイル, 検出器視野の GTI ファイルを読み込み、全ての GTI ファイルに共通する時間帯が書かれた最終 GTI ファイルを出力する。ここで得られる最終 GTI ファイルは、以降の科学プロダクツを生成するプロセスで使用する。
xselect	天体周辺のイベントファイル, 最終 GTI ファイルを読み込み、生プロダクツ (光度曲線, エネルギースペクトル), および最終プロダクツ (イメージ) を生成する。
lcmath	生プロダクツの光度曲線は、カメラ (§1 を参照) ごとに得られるので、lcmath を用いて足し合わせる。
mathpha	生プロダクツのエネルギースペクトルは、カメラごとに得られるので、mathpha を用いて足し合わせる。

SSC の科学プロダクツを作成できる, Perl スクリプトである “mxproduct” を提供する。それによって、エンドユーザーが行うべきことは簡素化され、以下の 3 つのステップのみである。

- 1) HEASARC/NASA からソフトウェア (HEAsoft), DARTS/JAXA もしくは HEASARC/NASA からアーカイブデータを取得する。
- 2) mxproduct に天体座標と観測期間を与えて実行する (オプションにより個別の要求に対応)。
- 3) 生成された科学プロダクツを用いて科学解析を行う。

エンドユーザーが用意するデータ、および mxproduct で実行されるプロセスを図 2 に示す。図 2 における各々のソフトウェアの動作内容は表 2 と表 3 に、各々のプロセスの動作内容は表 4 に示す。プロセスの流れは GSC と SSC で共通である。図 2 に記載されている “MAXI ソフトウェア” は、HEAsoft に組み込まれ、HEASARC から公開される予定である。したがって、データアーカイブが公開された際には、エンドユーザーが用意すべきソフトウェアは HEAsoft のみである。

ユーザーは、領域イベントファイル、Auxil ファイルが置かれているディレクトリにおいて mxproduct を実行する。引数は、天体座標 (赤経・赤緯)、GSC と SSC のエネルギーバンドが書かれたテキストファイル、解析したい期間の開始時刻と終了時刻の Modified Julian Date (エンドユーザーの GTI ファイル) である。その結果、有効面積が補正されていないイメージ、有効面積が補正された光度曲線とエネルギースペクトル、エネルギー応答関数ファイルが出力される。

上記に示した引数だけで最低限必要な科学プロダクツは得られるが、より高度な解析を行いたいエ

表 4: mxproduct における科学プロダクツの作成プロセスの動作内容

科学プロダクツ名	動作内容
光度曲線	mxgscandata (GSC 用) もしくは mxsscandata (SSC 用) で、最終 GTI ファイル、mxscancur の出力ファイル、天体周辺のイベントファイルを読み込み、時刻ごとの検出器有効面積を計算すると共に、光度曲線を作成する。その後、lcmath を用いて、検出器有効面積の補正を行い、使用できる全てのカメラについて足し合わせて、最終プロダクツである有効面積が補正された光度曲線を出力する。
エネルギースペクトル	xselect を用いて、天体周辺のイベントファイルと最終 GTI ファイルから、エネルギースペクトルを作成する。mathpha を用いて、使用できる全てのカメラについて足し合わせる。ここで得られるエネルギースペクトルは、有効面積の補正がされていない。mxgrmfgen (GSC 用) もしくは mxsrmfgen (SSC 用) を実行する際の引数に、前述のエネルギースペクトルを与えることで、最終プロダクツである有効面積が補正されたエネルギースペクトルが出力される。
エネルギー応答関数	mxgtiwwmap (MAXI に固有のソフトウェア) へ、Auxil ファイルと mxscancur の出力ファイルを読み込ませて、検出器応答関数ファイル (時系列の有効面積、積分した観測時間マップ) を作成する。mxgrmfgen (GSC 用) もしくは mxsrmfgen (SSC 用) を用いて、検出器応答関数ファイルから最終プロダクツであるエネルギー応答関数ファイルを作成する。

ンドユーザーのために、様々なオプションを引数で与えることができる。また、オプションを与えることで不要なプロセスを省略できる。

3 現在の状況と今後の計画

MAXI のデータアーカイブ開発について、現在の状況と今後の計画を解説する。全世界の科学者 (エンドユーザー) による MAXI のデータの利用を促進するため、MAXI 固有のソフトウェアを HEASoft へ組み込むための整備、およびデータアーカイブの公開を進めている。2015 年度末を目処に試験公開する予定であり、エンドユーザーへフィードバックを求めて、検討を行った後に、完全公開を行う予定である。完全性が保障されている観測期間についてはプロダクションアーカイブとして公開し、新しいデータはリアルタイムアーカイブとして公開する。較正において有意な改善があった場合は、MAXI データアーカイブチームで再較正を行い、データアーカイブの第 2 バージョンとして公開する予定である。Web から科学プロダクツ (イメージ、エネルギースペクトル、光度曲線、エネルギー応答関数) を得られるオンデマンド機能^{5)*6} を、UDON2 に追加する。現在は、オンデマンド機能を RIKEN で公開しており、これを UDON2^{*7} へ取り込み、JUDO2 から呼び出せるようにする予定である。また、JUDO2 からは MAXI のリアルタイム全天画像を公開する予定である。

MAXI データアーカイブの公開により、MAXI データの利用促進、科学的成果の最大化、査読論文の増加を期待している。また、X 線分野の科学者だけでなく、他波長の科学者による利用が進むことを期待している。

4 データアーカイブの開発に基づく教訓と指針

MAXI のデータアーカイブの開発を通して学んだ教訓を交えて、今後の宇宙機の開発において、ミッションの大小に関わらず、重要となると考えている指針を示す。

^{*6} <http://maxi.riken.jp/mxondem>

^{*7} <http://www.darts.isas.jaxa.jp/astro/udon.html>

指針 1: データアーカイブの開発はミッションの早い段階で検討・設計すること

当初はデータアーカイブの公開が計画されておらず、後期運用（2012年11月～、§1を参照）になってから本格的に開始したため、ソフトウェアの様々な部分で作り直しが生じた。宇宙機が取得したデータのアーカイブ化は、今後、ほとんどの場合において必要となると予想されるので、ミッションの早い段階からデータアーカイブの開発を視野に入れることが重要である。ただし、ミッションの開発を進めながら、データアーカイブ開発を行うことは難しいので、データアーカイブの開発に専念する人材を確保することが重要である。

指針 2: データは一つの機関で一元管理すること、システム構造を文書化・更新すること

MAXIDB はきちんと管理されていた一方で、MAXIDB から作成された Auxil ファイルや時系列イベントファイルは、置き場所が変更される場合があった。開発においては、システム構造を変更したい場合があるが、そのまま大規模化および複雑化すると、誰も全容を把握できなくなる。早い段階で十分な議論を行ってシステム構造を決めて、仮に変更の必要性が生じた場合はその妥当性を検討した上でチーム全体で合意を行うと共に、きちんと文書に反映する必要がある。

指針 3: 共有できる形で文書化を徹底すること

「誰かの頭の中」にだけある知識が多くあり、情報収集に時間を費やした。慣れていない文書化は大変な作業であり、時間がかかる場合があるが、ミッションを効率的に進める上で重要である。

指針 4: ソフトウェア開発の管理・情報共有を徹底すること

MAXI の場合、様々なソフトウェアパッケージがあったが、必ずしも相互に足並みを揃えて開発されていなかった。科学解析の効率化や手順の明瞭化を追い求めるうちに、情報共有が行われず、関数名が変更されたり、メインバージョンが分岐して開発されていたり、同様なソフトウェアが複数作成されていたりした。すぐれたデータアーカイブの公開を実現するためには、決定権を有する責任者、および実働する開発者を、チームにおいて定義して、その責任者と開発者が十分に議論した上で、ソフトウェア開発を進めることが重要である。

参考文献

- 1) Matsuoka, M., et al., The MAXI Mission on the ISS: Science and Instruments for Monitoring All-Sky X-Ray Images, PASJ, 61, 2009, 999.0906.0631
- 2) Mihara, T., et al., The Gas slit camera, GSC, onboard MAXI, in Astrophysics with All-Sky X-Ray Observations, edited by N. Kawai, T. Mihara, M. Kohama, & M. Suzuki, 2009, 14
- 3) Tomida, H. et al., Solid-state slit camera (SSC) onboard MAXI, in Astrophysics with All-Sky X-Ray Observations, edited by N. Kawai, T. Mihara, M. Kohama, & M. Suzuki, 2009, 18
- 4) Górski, K. M. et al. HEALPIX - a Framework for High Resolution Discretization, and Fast Analysis of Data Distributed on the Sphere, ApJ 622, 759, 2005
- 5) 中平聡志ほか, MAXI/GSC による全天観測データ公開システムの開発, 宇宙科学情報解析論文誌, 第二号, 2013, 29 (<https://repository.exst.jaxa.jp/dspace/handle/a-is/16827>)

JavaFX-based iUgonet Data Analysis Software (JudasFX)の プロトタイプ開発

小山 幸伸^{*1}, 佐藤 由佳^{*2} 中野 慎也^{*3} 八木 学^{*4} 田中 良昌^{*2} 阿部 修司^{*5}
能勢 正仁^{*6} 蔵川 圭^{*7} 池田 大輔^{*8} 梅村 宜生^{*9} 新堀 淳樹^{*10} 上野 悟^{*11}

Prototype Development of the JavaFX-based iUgonet Data Analysis Software (JudasFX)

Yukinobu KOYAMA^{*1}, Yuka SATO^{*2}, Shinya NAKANO^{*3}, Manabu YAGI^{*4},
Yoshimasa TANAKA^{*2}, Shuji ABE^{*5}, Masahito NOSE^{*6}, Kei KURAKAWA^{*7},
Daisuke IKEDA^{*8}, Norio UMEMURA^{*9}, Atsuki SHINBORI^{*10}, and Satoru UeNo^{*11}

Abstract

By the Inter-university Upper atmosphere Global Observation NETwork (IUGONET) project which has started in 2009, (a)various long-term upper atmospheric data from 1800 's to present were released to the public, (b)improvement of distributed management data search by domain metadata of upper atmospheric research field, (c)improvement of environment for the data usage by the developing and releasing the data visualization and analysis software that deal with various data formats, were done for domain researchers of upper atmospheric research field. However, use of expensive commercial software is presupposed in the data visualization and analysis software in upper atmospheric research field. The available function is restricted to the user who doesn't have the license. Then we developed the prototype of free data visualization and analysis software which is called JudasFX in order to expand the potential data user such as researcher in developing countries and neighbor fields, data scientists, and citizens in general to promote interdisciplinary study, data-intensive science, and citizen science, respectively.

In this paper, we explain details of the software, and show visualization and analysis of geomagnetic Dst-index as a case example.

Keywords: Upper Atmosphere, Visualization, Analysis, Open Source Software, Data-intensive Science, Open Science

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

^{*1} 情報・システム研究機構, 新領域融合研究センター

(Transdisciplinary Research Integration Center, Research Organization of Information and Systems)

^{*2} 情報・システム研究機構, 国立極地研究所

(National Institute of Polar Research, Research Organization of Information and Systems)

^{*3} 情報・システム研究機構, 統計数理研究所

(The Institute of Statistical Mathematics, Research Organization of Information and Systems)

^{*4} 東北大学惑星プラズマ・大気研究センター (Planetary Plasma and Atmospheric Research Center, Tohoku University)

^{*5} 九州大学国際宇宙天気科学・教育センター

(International Center for Space Weather Science and Education, Kyushu University)

^{*6} 京都大学大学院理学研究科附属地磁気世界資料解析センター

(Data Analysis Center for Geomagnetism and Space Magnetism, Graduate School of Science, Kyoto University)

^{*7} 情報・システム研究機構, 国立情報学研究所

(National Institute of Informatics, Research Organization of Information and Systems)

^{*8} 九州大学大学院システム情報科学研究院

(Graduate School of Information Science and Electrical Engineering, Kyushu University)

^{*9} 名古屋大学宇宙地球環境研究所 (Institute for Space-Earth Environment Research, Nagoya University)

^{*10} 京都大学生存圏研究所 (Research Institute for Sustainable Humanosphere, Kyoto University)

^{*11} 京都大学大学院理学研究科附属天文台 (Kwasan and Hida Observatories, Graduate School of Science, Kyoto University)

概要

「超高層大気長期変動の全球地上ネットワーク観測研究」プロジェクト(IUGONET, 平成21年度～)によって, (a)1800年代から現在に至る長期で様々な超高層大気の地上観測データ公開, (b)分散管理されている超高層物理学のデータのドメイン・メタデータ共有による検索性の向上, そして(c)様々なデータ形式を取り扱う統合データ可視化・解析ソフトウェアの開発と公開によるデータ利用環境の向上が, 超高層物理学のドメイン研究者向けに行われてきた. しかしながら, 超高層物理学のドメイン研究者向けの多くの統合データ解析ソフトウェアは, 高価な商用ソフトウェアの利用を前提としており, ライセンスが無ければ, 利用可能な機能が制限される. そこで我々は, データ中心科学, 学際的研究, シチズンサイエンスの推進に先立って, データ・サイエンティスト, 開発途上国の研究者, 天文や気象等の隣接分野の研究者, さらには一般市民にデータ利用可能性を広げるために, フリーのデータ可視化・解析ソフトウェア JudasFX のプロトタイプを開発した. 本論文では, そのソフトウェアの詳細を説明し, 地磁気 Dst 指数を可視化・解析した事例を示す.

キーワード: 超高層大気, 可視化, 解析, オープンソース・ソフトウェア, データ中心科学, オープン・サイエンス

1. はじめに

従来の1. 実験 (帰納型), 2. 理論 (演繹型), そして3. 数値計算 (演繹型) に基づいた科学データの増大, いわゆるビッグ・データ問題の顕在化と, 第4の科学的パラダイムであるデータ中心科学 (帰納型) の到来が T. Hey らによって示唆された¹⁾. 図 1に示す通り, 学術情報は Literature (文献), Intermediate Data (中間データ), Published Data (出版データ) で大別されるが, 2004年頃から, 世界の論文数の増加率が上がり²⁾, 他方でセンサーやコンピューターから生成されるデータは, それ以上に日々爆発的に増加しているのが現状である³⁾. 1665年以降, 自然言語を中心として記述・継承されてきた学術情報であるが, データ中心科学時代における学術情報基盤の課題として, 科学論文の論拠をデータによって明すための, 文献, 中間データ, 出版データの各層間のインターネット上での結合と, 科学的知見を速やかに得るための, 大量データ処理が挙げられる.

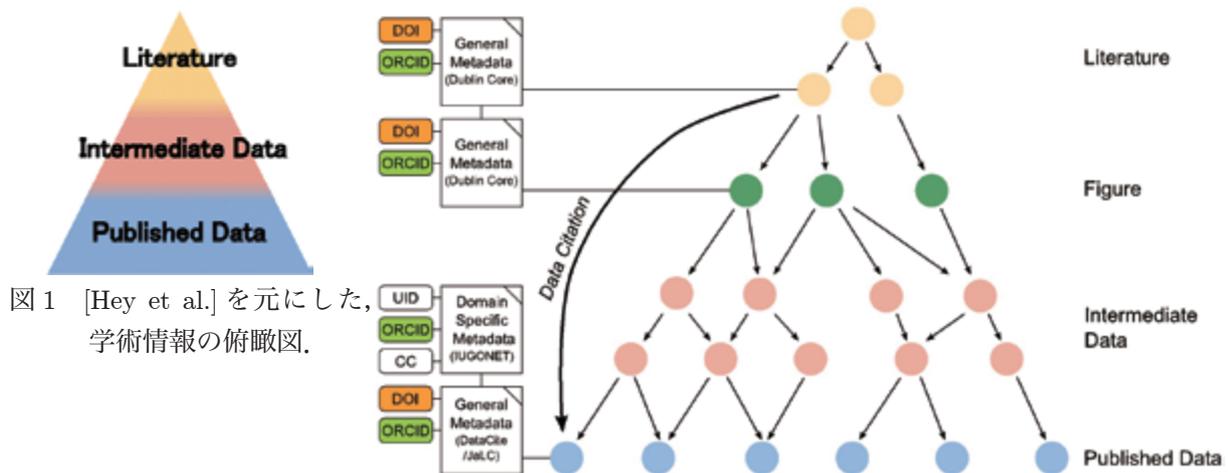


図 1 [Hey et al.] を元にした, 学術情報の俯瞰図.

図 2 超高層物理学分野における, 科学論文の論拠となる各種の学術情報とそれらの関係性.

2. 超高層物理学における学術情報基盤

超高層物理学においては、下層大気の温暖化に伴う超高層大気の寒冷化に代表される、地球規模の諸現象解明がますます重要となっており、ビッグデータ解析がその鍵となる。そして、ビッグデータを取り扱った分野横断型研究を、円滑に進めるための基盤整備こそが喫緊の課題である。超高層物理学分野における、科学論文の論拠となる各種の学術情報とそれらの関係性を図 2 に示す。Digital Object Identifier (DOI) は永続的な識別子、Unique Identifier (UID) は後述する超高層物理学の IUGONET プロジェクト内で通用するデータ識別子、Open Researcher and Contributor Identifier (ORCID)⁴⁾ は研究者識別子、Creative Commons (CC) はコンテンツ、ここではデータの利用規約を定めるライセンスである。データ公開や、データに DOI を付与して一般的なメタデータを付けるデータ出版⁵⁾、DOI を介して文献とデータを結びつけるデータ引用等の種々の取り組みによって⁶⁾⁷⁾⁸⁾⁹⁾、文献からデータへの到達可能性が高まりつつある。さらには、「超高層大気長期変動の全球地上ネットワーク観測研究」プロジェクト (IUGONET: Inter-university Upper Atmosphere Global Observation NETWORK) によって、データの科学的な解釈に必要なドメイン・メタデータが、機械可読形式で取得可能になった¹⁰⁾¹¹⁾¹²⁾。データから如何に迅速かつ効率的に文献にするかという課題に先立って、次に取り組むべきは、データの可視化と、近年課題となっているビッグデータから中間データへの導出である。ビッグデータは、「量」、「速度」、「種類」の3種の組み合わせで特徴づけられるが、超高層物理学のデータでとりわけ特徴的なのは「種類」であり、取り扱う物理量の多様性と、様々なデータ形式の乱立が問題である。

Linked Data¹³⁾ に例示される様に、誰でも入手可能で汎用的な可視化・解析ソフトウェアとして取り扱うために、データ形式を極力統一する方向性で議論されるのが通常であるが、超高層物理学においては、各研究コミュニティ内で、長期に渡って利用されてきたデータ形式の変更はもとより、現状のデータ公開に加えて、Resource Description Framework (RDF) 形式等のコミュニティにとってなじみの無い新たなデータ形式でデータ提供することにも抵抗があるため、当面の間、データ形式の集約は困難である。この理由から、データ形式の差異を吸収するデータ解析ソフトウェアがドメイン研究者向けにいくつか開発されてきた。この実装のひとつを図 3 に示す。IUGONET が作成した地上観測データ用の iUgonet Data Analysis Software (UDAS)¹⁴⁾¹⁵⁾¹⁶⁾、THEMIS 衛星のグループが作成した衛星観測データ用の Themis Data Analysis Software suite (TDAS)¹⁷⁾、そしてそれらを統合した、様々な観測データを統一的に取り扱う統合データ解析ソフトウェアである、Space Physics Environment Data Analysis Software (SPEDAS)¹⁸⁾ が、商用ソフトウェアの Interactive Data Language (IDL) 上で開発されてきた。ドメイン研究者向け可視化・解析ソフトウェアとして一定の成功を収めた SPEDAS であるが、ドメイン研究者以外がそれを利用する場合、次の2つの問題に直面する。

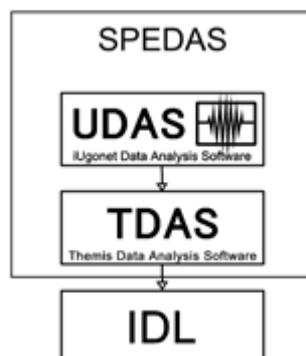


図 3 SPEDAS の構成図。

ライセンス問題： 商用ソフトウェアの IDL を基盤ソフトウェアとしたことによるライセンス問題を抱える。IDL は、医学、天文学、地球科学の一部の普及に留まっており、ドメイン外の人々が誰でも利用できる環境は整備されていない。ライセンスが不要な SPEDAS Virtual Machine も用意されているが、Command-line Interface (CLI) が使えない限定機能版であるため、解析手順のスク립ト化や、ユーザーの自由な発想に基づいた解析プログラムの作成が出来ず、ライセンスを所有している一部の限られたソフトウェア開発者によって、あらかじめ提供されている定型処理を、Graphical User Interface (GUI) を用いて逐次的に実行するほかない。また、ライセンス数と同数のプロセスしか起動出来ないため、多数のプロセッサを同時に用いることも困難である。これらの理由から、ビッグデータを用いた解析用途には不十分である。さらには、ソフトウェア開発にもライセンスが必要であるため、開発コミュニティ規模がライセンス数の制約を受け、結果としてソフトウェア資産が充実しない。ドメイン内であっても商用ライセンスの購入が困難な開発途上国の研究者、天文や気象などの隣接分野の研究者、データ・サイエンティスト、さらには一般市民らといったドメイン外の人々が、データ利用およびソフトウェア開発できる環境が整っていない。

相互運用性の欠如： 欧州非干渉散乱レーダー(EISCAT)¹⁹⁾においては、IDL 用と MATLAB 用のライブラリが、別グループにより開発されており、それらは相互運用出来ない。MAGDAS²⁰⁾地磁気データにおいても、同様の状況である。また、超高層物理学の SPEDAS と、太陽分野の SolarSoft²¹⁾は、互いに相互運用の要望があるにも関わらず、また双方とも IDL を基盤ソフトウェアとして使っているにも関わらず、IDL に名前空間の概念が無いために、衝突して共存出来ない。2015年11月現在、最新の SPEDAS と SolarSoft には、195個の同名プロシージャが存在する。この内、中身が完全に一致するプロシージャは57個で、残りの148個は中身が異なる。この中には、版違いなどの比較的軽微な差異の場合もあるが、例えば、SPEDAS における array_concat 「プロシージャ」と SolarSoft における array_concat 「関数」に代表される様に名前の衝突が生じている場合があるため、それらを同時に使用することが出来ない。この様な理由から、一方のライブラリ上で、既に実装されている機能を他方で再実装する事例が見られる。以上で例示された様に、相互運用出来ないデータ解析ソフトウェアの乱立が顕在化してきた。現状と、次節で詳細説明する本取り組みの模式図を図 4 に表す。

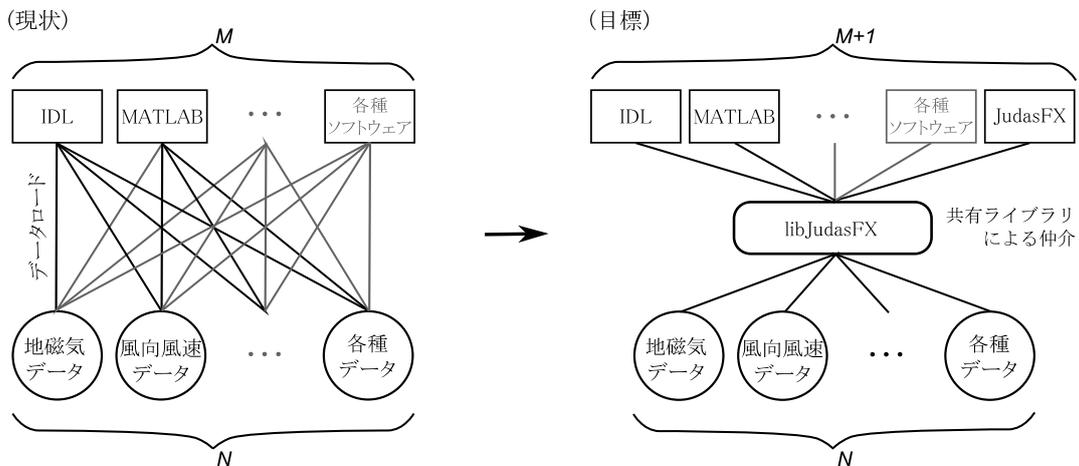


図 4 ソフトウェア開発の方針。

3. JavaFX-based iUgonet Data Analysis Software (JudasFX)の特徴

ドメイン外の人々がデータを利用する際に直面する様々な問題を解消するために、ライセンスフリーで相互運用性を考慮した、データ可視化・解析ソフトウェア JavaFX-based iUgonet Data Analysis Software (JudasFX)のプロトタイプを開発した。以下にその特徴について記述する。

ライセンスフリー： JudasFX のプロトタイプの全ソースコードは GPL v2.0ライセンスに従って、Github 上で公開している²²⁾。開発途上である現在はソースコードの公開のみであるが、将来的には Java 中間コードも公開・配布することを計画している。これにより、もし最新プログラムがリリースされていれば、Java Web Start の機能によって、ウェブサーバーから JudasFX を自動ダウンロードし、もし更新されていなければ、以前にダウンロードしたローカルに存在する実行ファイルを即座に利用することが可能となる。JudasFX はライセンス数の制約が無いため、多数のプロセッサを同時に用いたビッグデータ解析が可能である。JudasFX は、無料で手に入る Java Runtime Environment (JRE)環境上で動作する。仮に JRE が用意されていない環境下において JudasFX の起動を試みた場合、JRE のインストールが促され、クリックすることによって JRE がインストールされ、JudasFX の動作環境が直ちに整備される。ソフトウェア実行環境のみならず、開発環境である Java Development Kit, さらには開発を支援するための 統合開発環境 (IDE: Integrated Development Environment)である NetBeans や Eclipse も無料で入手可能である。JRE, JDK, NetBeans, Eclipse は、いずれも Windows, Mac OS X, Linux 用が用意されている。

相互運用性への配慮： 図 5に、JudasFX の構成図を示す。JudasFX は、Java Virtual Machine (Java VM)を介し、Windows, Mac OS X, Linux, Solaris のいずれの環境でも動作する。相互運用性への配慮から、スタンドアロンでの利用に留まらず、他の基盤ソフトウェアやプログラム言語からの利用も想定している。例えば、IDL や MATLAB は、Java ブリッジを備えているため、外部 Java ライブラリを利用可能である。そのため、既存の IDL や MATLAB ユーザーは、Java VM を介して JudasFX のプラグイン部分である libJudasFX を利用し、データのダウンロード機能等を使うことが出来る。オブジェクト指向言語である Java 言語を用いて記述されている本ライブラリは、名前空間で切り分けられているので相互運用性が高く、他の言語からの利用も容易である。

この様に、ドメイン研究者を中心として一定数存在する商用ソフトウェアのユーザーに対して相互運用性の面で配慮しつつ、新たに開発途上国の研究者、隣接分野の研究者、データ・サイエンティスト、さらには一般市民を中心として JudasFX のユーザーを増やし、全体的な開発リソースの集約とデータ利用環境の向上を狙う。

オフラインでの利用が可能： ネットワーク事情が良くない地域にも配慮し、ひとたび JudasFX とデータをダウンロードしておけば、インターネットに常時接続していなくてもオフラインで利用可能にする点においては、Web ベースの可視化・解析ソフトウェアに対して優位性がある。他方で、Web ベースの可視化・解析ソフトウェアは、サーバー上のソフトウェアを更新するだけで全利用者が最新のソフトウェアを使える点で優れている。そこで、サーバー上に置かれた Java アプリケーションを自動ダウンロード、自動インストール、自動アップデートして、実行可能な仕組みである Java Web Start を利用することにより、JudasFX は最新ソフトウェアの再配布性を高める。

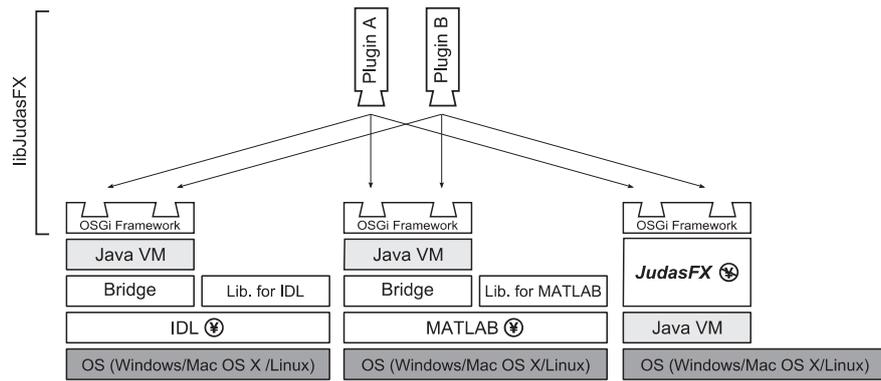


図 5 JudasFX の構成図。

図 6に、JudasFX の主要クラスを示す。データファイルのダウンロードを行い、メタデータを取扱う APlot 抽象クラスを定義した。さらには、超高層物理学の多様な物理量をある程度統一的に取り扱うために、時系列データを扱う TPlot、地図上のデータを扱う MPlot、画像データを扱う IPlot、スキャッタープロットのデータを取り扱う XYPlot の4種の抽象クラスを定義した。実際にデータを取り扱うには、抽象クラスを継承した具象クラスが必要となる。例えば、地磁気擾乱を表す指数のひとつである地磁気 Dst 指数を取り扱う DstIndex 具象クラスは、時系列データを扱う TPlot 抽象クラスを継承して実装されている。なお、親クラスである TPlot 抽象クラスのメンバー変数に時系列データが代入されるように、DstIndex 具象クラスは実装されている。この様に、親クラスのメンバー変数に代入するまでを受け持つ個別の具象クラスが対象データセット数だけ実装される。子クラスは継承によって親クラスのメソッドも継承するため、例えばどの具象クラスにおいても file_http_copy()メソッドがデータファイルをダウンロードするコマンドである。多様なデータ種を用いた統合データ解析が必要な超高層物理学においては、このような統一的なデータアクセス手法が重要となる。なお、TPlot クラスと XYPlot クラスは Java の基本クラス、MPlot クラスでは OSGeo による GeoTools²³⁾、IPlot クラスでは OpenCV²⁶⁾を各々再利用しており、極力既存のライブラリを再利用するよう留意している。

現在は、JudasFX の開発の以前に進めていた、Java, Swing, そして JFreeChart を用いて記述した約20個の古い実装、例えば京大地磁気センターが公開している地磁気データ、京都大学生存圏研究所が公開している MU レーダーによる風速データ、国立極地研究所や名古屋大学が公開している SuperDARN レーダーデータを取り扱うクラス等を、順次 JavaFX 用書き直している段階である。拡張性を考慮するため、各種データセットを取り扱うクラスは、OSGi フレームワークの実装である、Apache Felix を用いてプラグイン形式で追加される。将来構想として、TPlot, MPlot, IPlot, XYPlot の各親クラスにおいて、Linked Open Data の形式で、メンバー変数を出力する付加機能を実装することを検討しており、本プログラムを介さずに、静的にかつ統一化されたデータファイルにアクセスしたいという潜在的ユーザーにも対応したいと考える。

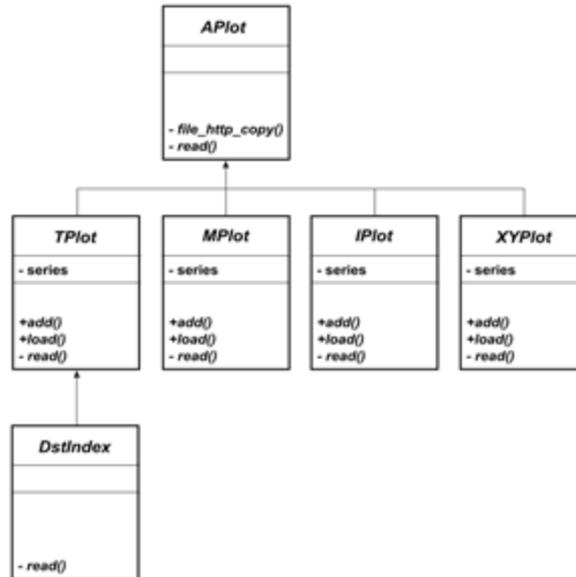


図6 JudasFXの主要クラス図.

3.2 堅牢性の向上

図7にModel-View-Controllerの関連図を示す。従来のJava言語を用いたGUIプログラミングでは、デザインパターンのひとつであるModel-View-Controller²⁵⁾のかかなりの部分がひとつのJavaソースコード内に記述され、保守性が著しく悪かった。しかしながら、JavaFXをベースにすることによって、ボタン等の各部品配置情報を保持するViewの部分をJava言語と切り離して、XMLベースのマークアップ言語であるFXML形式で記述することが可能になり、保守性が著しく上がった。また、Modelに関しては、前出のとおり、超高層物理学分野においては、データ形式が集約されていないため、個々の取り扱いデータセット数分の個別の実装が必要となり、ソースコードが冗長化しがちである。そこで、共通部分は極力親クラスで実装することで、ソースコードの共通化を心がけた。また、「Model」と「Controller」を別のJavaソースコードに切り分けて記述することが容易になったので、過去に作成したJava, Swing, そしてJFreeChartを用いたプログラムよりも保守性が高まった。

以上のように、切り分けてコーディングしやすくなったソースコードの堅牢性をさらに向上させるために、Modelに関してはJUnitを用いた自動の単体テストをソースコードの改変ごとに行い、ViewならびにControllerのボタン操作などのGUI周辺を含んだテストは、JUnitとTestFXを用いた自動の統合テストをソースコードの改変ごとに行っている。

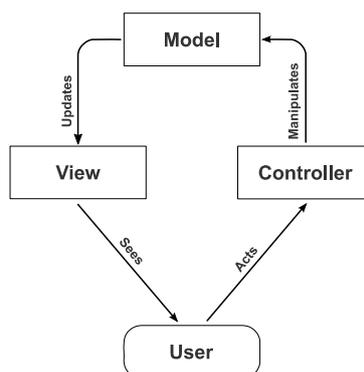


図7 Model-View-Controllerの関連図.

3.3 インターフェースおよび数値計算ライブラリ

図 8 の左図に、JudaxFX のスクリーンショットを示す。SPEDAS を利用している超高層大気の研究にとって馴染みがあるという理由から、JudaxFX の初版は SPEDAS ライクな GUI を作成した。フリーの JavaFX 用の GUI 開発アプリケーションである SceneBuilder を用いて、グラフィカルに可視化・解析ソフトウェアの Graphical User Interface (GUI) を作成した。

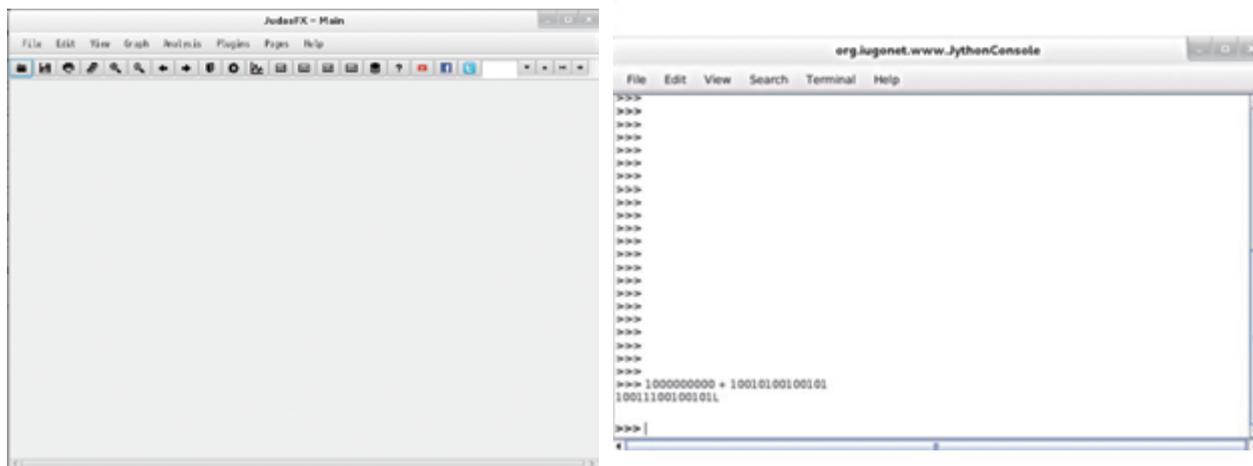


図 8 JudaxFX のスクリーンショット (左) と CLI 窓 (右)。

図 8 の右図に、CLI 窓を示す。CLI には、Jython 環境を用意した。データサイエンティストが利用するデファクトスタンダードのツールのひとつである Python を利用したかったが、他方で Java 実行環境以外の如何なる依存ソフトウェアの事前インストールも、JudaxFX 利用の前提にしたくなかった。この折衷案として、Python の Java で書かれた実装である Jython を CLI として選んだ。JudaxFX 本体と同様に、Jython.jar と JyConsole.jar で構成される Jython 環境もまた Java Web Start によってクライアントサイドへ送信されるため、ローカルコンピュータに Jython を事前インストールする必要が無い。

数値計算環境については、数値計算ライブラリである Apache Commons Math²⁴⁾を利用可能である。将来的には、Python と合わせて利用される NumPy や SciPy を利用したいので、ネイティブ Python の拡張を Jython で使うための拡張である Jython Native Interface (JyNI)を利用し、Jython 上で Numpy や SciPy などを利用するテストを行っており、データ・サイエンス環境を構築する予定である。

3.4 実行例

ここでは、JudasFX の実行例を示す。図 9の左図は、LoadData 窓であり、プロットするデータセット種、日時、観測所名等を選択する。これらの選択後、バックエンドでは下記の動作が行われる。

1. IUGONET メタデータ・データベースが、外部アプリケーションとの通信用に用意している OpenSearch インターフェースに対して、JudasFX が HTTP 経由でメタデータ検索クエリーを実行する。
2. データセットのメタデータである NumericalData リソースタイプと、各データファイルのメタデータである Granule リソースタイプを、JudasFX が機械的に解釈する。これによって、データファイルの所在情報である AccessURL と、座標軸の物理量などが記載された RenderingHint を得る。
3. AccessURL に記載されたデータファイルを自動的にダウンロードする。その後、個別データを扱うためのクラスが持つメンバー変数にロードする。

この後、さらにプロットの指令を行うことにより、図 9の右図のプロットが作成される。ここでは、1984年10月の地磁気 Dst 指数の1時間(確定値)のプロットを図 9の右図に示した。比較のために、データ提供元である京都大学大学院理学研究科附属地磁気世界資料解析センターで公開されている、1984年10月の地磁気 Dst 指数の1時間値(確定値)のプロット画像を図 10に示す。

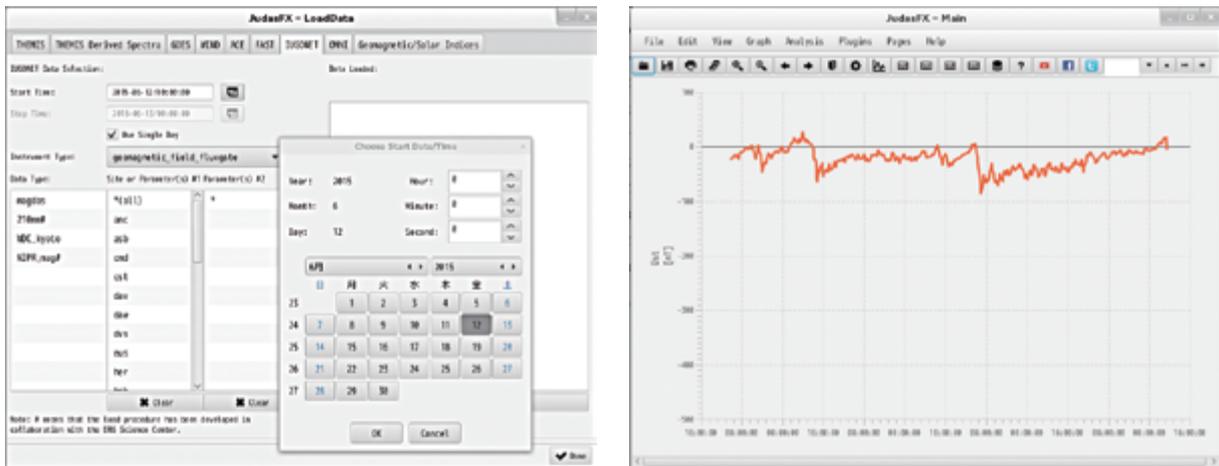


図 9 LoadData窓(左)と1984年10月の地磁気Dst指数の1時間値(右)。

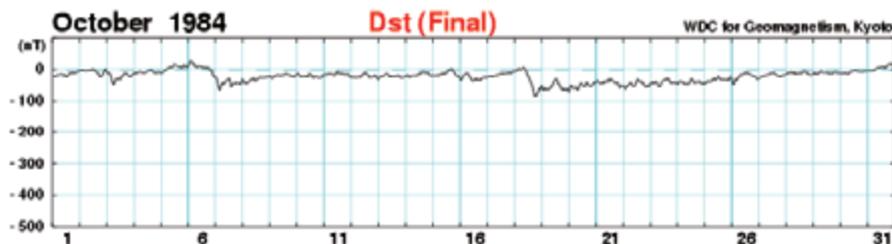


図 10 京都大学大学院附属地磁気世界資料解析センターにおいて公開されている、1984年10月の地磁気 Dst 指数の1時間値(確定値)のプロット画像。

次に、データ・プロセッシング窓のスクリーンショットを図 11の左図に示す。TPlot クラスを継承した各データセットを取り扱う個別クラスが保持するデータ変数に対して、GUI のボタンにあらかじめ仕込んである各種データ処理を行うことが出来る。ここでは、図 9の右図で得られた地磁気 Dst 指数に対して、24時間移動平均値を行ったものを図 11の右図に示す。移動平均のメソッド内部において、Apache Commons Math を呼び出している。GUI による簡素なデータ処理のみならず、Jython と Apache Commons Math を用いた CLI によるデータ処理も利用可能である。以上のとおり、商用ライセンス不要でデータの可視化・解析が可能となった。

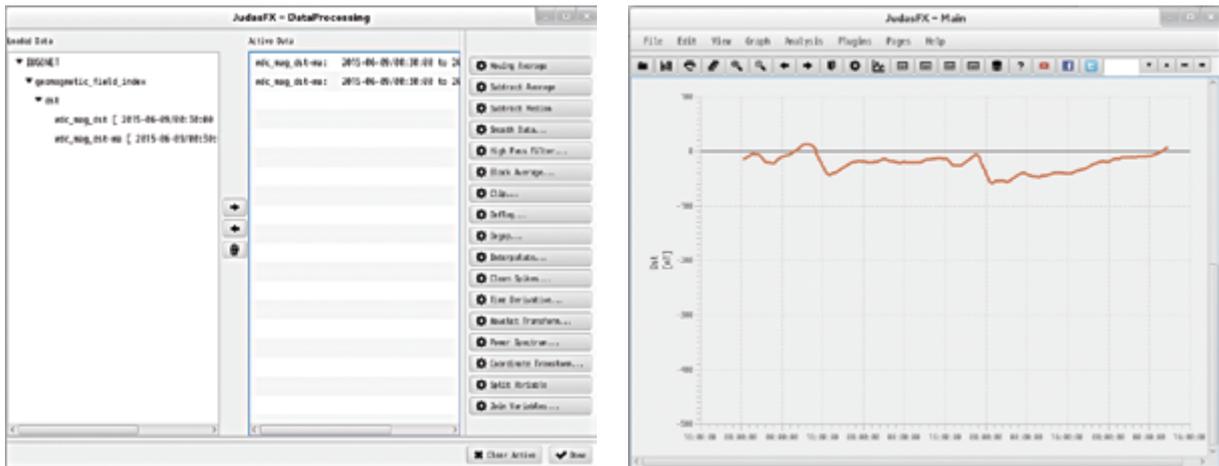


図 11 DataProcessing 窓(左)と1984年10月の地磁気 Dst 指数の1時間値の24時間移動平均値(右)。

4. まとめと今後について

本取り組みでは、超高層物理学におけるデータ利用環境向上のため、JavaFX ベースのデータ可視化・解析ソフトウェアである JudasFX のプロトタイプを作成した。途上国の研究者、隣接分野の研究者、データサイエンティスト、さらには地球環境に興味を抱く一般市民でも利用出来るように、単独で動く可視化・解析ソフトウェアを実装した。ライセンス数に縛られないため、多数のプロセッサを用いたビッグデータ解析が可能であり、過去にこの分野で無かったデータ中心科学を切り開くツールとして期待される。

今後、JudasFX を IUGONET メタデータ・データベースへ組み込むことにより、オンラインでの可視化機能を提供することを予定している。さらに、Japan Link Center によって進められている「研究データへの DOI 登録実験プロジェクト」²⁷⁾において生成される、データセットのランディングページから、IUGONET メタデータ・データベースへ誘導することにより、文献を起点としたデータの可視化までが円滑になると期待される。

謝辞

情報・システム研究機構新領域融合研究センターによる、融合研究シーズ探索の支援を受け、本研究を推進した。また、京都大学による、融合チーム研究プログラム(SPIRITS)の支援を受け、本研究を推進した。本論文では、京都大学大学院理学研究科附属地磁気世界資料解析センターの Web のスナップショットならびに、地磁気 Dst 指数を利用した。著者一同は、上記の全てに対して感謝する。

参考文献

- 1) TONY HEY, STEWART TANSLEY, AND KRISTIN TOLLE, The FOURTH PARADIGM: DATA-INTENSIVE SCIENCE DISCOVERY, MICROSOFT RESEARCH, 2009.
- 2) 阪 彩香, 桑原 輝隆, 科学研究のベンチマーキング2012 -論文分析でみる世界の研究活動の変化と日本の状況-, 文部科学省 科学技術政策研究所, 2013年3月.
- 3) 平成26年版 情報通信白書, 総務省, 平成26年.
- 4) 蔵川 圭, 武田 英明, 研究者識別子 ORCID の取り組み, 情報管理, doi:10.1241/johokanri.54.622, 2011, Vol. 54, No. 10.
- 5) 武田英明, 機関リポジトリと DOI ~研究データ流通を支える情報基盤とは~
http://www.nii.ac.jp/sparc/event/2014/pdf/20150309_4.pdf.
- 6) [http://onlinelibrary.wiley.com/journal/10.1002/\(ISSN\)2049-6060](http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)2049-6060)
- 7) <http://www.nature.com/sdata/>.
- 8) <http://www.earth-system-science-data.net/>.
- 9) <http://www.journals.elsevier.com/data-in-brief/>.
- 10) 林 寛生, 小山 幸伸, 堀 智昭, 田中 良昌, 新堀 淳樹, 鍵谷 将人, 阿部 修司, 河野 貴久, 吉田 大紀, 上野 悟, 金田 直樹, 米田 瑞生, 田所 裕康, 元場 哲郎, 大学間連携プロジェクト「超高層大気長期変動の全球地上ネットワーク観測・研究」, 宇宙科学情報解析論文誌 第1号, JAXA-RR-11-007, 2012, 113-120.
- 11) Hayashi, H., Y. Koyama, T. Hori, Y. Tanaka, S. Abe, A. Shinbori, M. Kagitani, T. Kouno, D. Yoshida, S. UeNo, N. Kaneda, M. Yoneda, N. Umemura, H. Tadokoro, T. Motoba, and IUGONET project team, Inter-university Upper atmosphere Global Observation NETwork (IUGONET), Data Sci. J., 12, doi:10.2481/dsj.WDS-030, 2013, WDS179-WDS184.
- 12) 小山 幸伸, 河野 貴久, 堀 智昭, 阿部 修司, 吉田 大紀, 林 寛生, 田中 良昌, 新堀 淳樹, 上野 悟, 金田 直樹, 米田 瑞生, 元場 哲郎, 鍵谷 将人, 田所 裕康, 超高層物理学分野のためのメタデータ・データベースの開発, 宇宙科学情報解析論文誌 第1号, JAXA-RR-11-007, 2012, 99-104.
- 13) トム・ヒース, クリスチャン・バイツァー著, 武田 英明(監訳), 大向 一輝(訳), 加藤 文彦(訳), 嘉村 哲郎(訳), 亀田 堯宙(訳), 小出 誠二(訳), 深見 嘉明(訳), 松村 冬子(訳), 南 佳孝(訳), Linked Data: Web をグローバルなデータ空間にする仕組み, 近代科学社, ISBN-

- 13:978-4764904279, 2013.
- 14) 田中 良昌, 新堀 淳樹, 鍵谷 将人, 堀 智昭, 阿部 修司, 小山 幸伸, 林 寛生, 吉田 大紀, 河野 貴久, 上野 悟, 金田 直樹, 米田 瑞生, 田所 裕康, 元場 哲郎, 三好 由純, 関 華奈子, 宮下 幸長, 瀬川 朋紀, 小川 泰信, IUGONET 解析ソフトウェアの開発, 宇宙科学情報解析論文誌 第1号, JAXA-RR-11-007, 2012, 91-98.
 - 15) 田中 良昌, 新堀 淳樹, 梅村 宜生, 堀 智昭, 阿部 修司, 小山 幸伸, 林 寛生, 上野 悟, 佐藤 由佳, 谷田 貝 亜紀代, 小川 泰信, 三好 由純, 関 華奈子, 宮下 幸長, 瀬川 朋紀, IUGONET 解析ソフトウェアの現状と今後の発展, 宇宙科学情報解析論文誌 第2号, JAXA-RR-12-006, 2013, 63-70.
 - 16) Tanaka, Y., A. Shinbori, T. Hori, Y. Koyama, S. Abe, N. Umemura, Y. Sato, M. Yagi, S. Ueno, A. Yatagai, Y. Ogawa, and Y. Miyoshi, Analysis software for upper atmospheric data developed by the IUGONET project and its application to polar science, *Adv. Polar Sci.*, 24, doi:10.3724/SP.J.1085.2013.00231, 2013, 231-240.
 - 17) <http://themis.ssl.berkeley.edu/software.shtml>.
 - 18) Hori, T., M. Miyashita, M. Miyoshi, K. Seki, T. Segawa, Y. M. Tanaka, K. Keika, M. Shoji, I. Shinohara, K. Shiokawa, Y. Otsuka, S. Abe, A. Yoshikawa, K. Yumoto, Y. Obana, N. Nishitani, A. S. Yukimatsu, T. Nagatsuma, M. Kunitake, K. Hosokawa, Y. Ogawa, K. T. Murata, M. Nose, H. Kawano, and T. Sakanoi, CDF data archive and integrated data analysis platform for ERG-related ground data developed by ERG Science Center (ERG-SC), 宇宙科学解析論文誌 第4号, 2015, 75-89.
 - 19) <https://www.eiscat.se/>.
 - 20) <http://magdas.serc.kyushu-u.ac.jp/>.
 - 21) <http://sohowww.nascom.nasa.gov/solarsoft/>.
 - 22) <https://github.com/koyamalmsteen/JudasFX/>.
 - 23) <http://geotools.org/>.
 - 24) <https://commons.apache.org/proper/commons-math/>.
 - 25) <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>.
 - 26) <http://opencv.org/>.
 - 27) 小林 賢, ジャパンリンクセンター「研究データへの登録実験プロジェクト」中間報告会, Vol. 58, No. 6, doi:10.1241/johokanri.58.485, 情報管理, 2015, 485-488.

地球惑星科学分野のクイックルックサービス

今井 弘二^{*1}, 海老沢 研^{*1}, 池田 大輔^{*2}, 北尾 大介^{*2}

Quick look service for geoscience

Koji Imai^{*1}, Ken Ebisawa^{*1}, Daisuke Ikeda^{*2}, Daisuke Kitao^{*2}

Abstract

It is important to progress cross-cutting researches in geoscience to understand how the various geophysical phenomena have influence on each other. However, it is not easy to check scientific data for geoscientist in other fields because each area has been developing independently. Therefore, we had built a new web service, C3 (Cross-Cutting Comparisons) for promotion of the data utilization. By the interactive interface, C3 reduces distances between the fields in geoscience and provides a quick look viewer. This article describes the system summary and features of the service.

Keywords: Geoscience, Web service, Quick look, Cross-cutting research

概要

太陽から人類生存圏までの空間における様々な現象が、どのように影響を及ぼし合っているのかを把握するためには、地球惑星科学における多分野のデータを用いた複合研究を進める必要がある。しかしながら、各分野は独自の進歩を遂げてきたために、それらのデータを確認することさえ容易ではない。そこで我々は、地球惑星科学分野におけるデータ利用を促進するためのウェブサービス C3 (Cross-Cutting Comparisons) の開発を進めている。そのインタラクティブな操作性によって異分野間の隔たりを緩和し、多様なデータを確認できるビューワを提供している。本稿では、そのシステムの概要とサービスの特徴について記述する。

キーワード：地球惑星科学，ウェブサービス，クイックルック，分野横断型研究

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

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

^{*2} 九州大学大学院システム情報科学 研究院・学府
(Graduate School of Information Science and Electrical Engineering, Kyushu University)

1 はじめに

地球温暖化やそれに伴う異常気象がますます問題視される中、地球の環境変化をもたらす様々な要因を把握することは、人類の最大課題の一つである。その環境変化の内、人類の活動に影響を与える気象や気候の変化は、主に人類の生存圏に要因があると考えられてきた。しかしながら、近年はそれよりも上層における微量分子の変動や、更には太陽活動の変化も様々な過程を経て下層大気に影響を及ぼすとの認識が深まってきている^{1,2)} (図1)。特に現在の太陽活動は、従来の予想と異なる振る舞いをしており、その変動がもたらす地球大気を構成する因子への影響に、より一層の関心が高まっている。

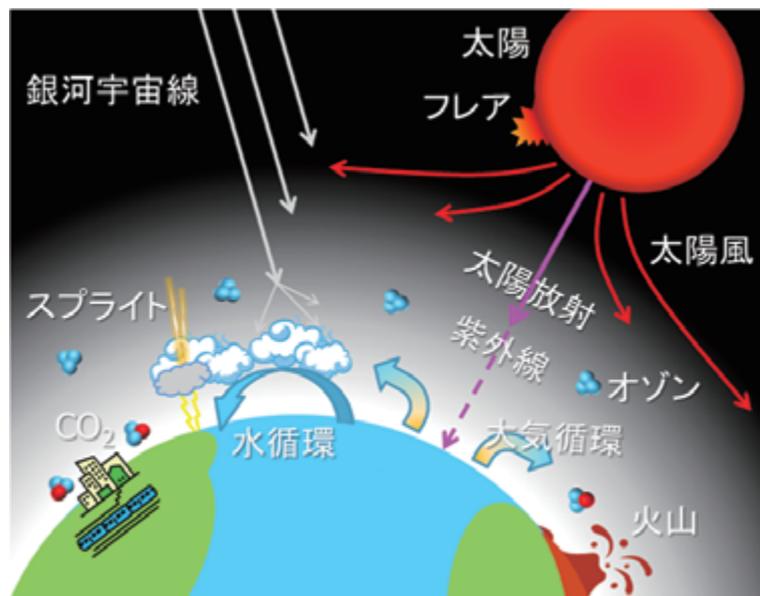


図1 太陽地球環境の模式図

一方、上層大気では太陽活動だけではなく、人類生存圏の諸現象（地震や雷放電、火山活動など）や、人為起因の大気汚染の影響も受けていることが示唆されている^{3,4,5,6)} (図1)。このように、太陽から人類生存圏までの空間（太陽地球環境）は様々な因子が複雑に絡み合っているため、その空間を一つの複合系として理解するためには、地球惑星科学における多分野（太陽物理学や大気科学、地質学など）の知識やデータを総動員した分野横断型の複合研究を進める必要がある（以下、特に断りがなければ地球惑星科学における分野横断型の研究を分野横断研究と略記する）。

太陽地球環境内で起きる突発現象や長期変動がもたらす影響を捉えるためには、その物理・化学素過程を様々な時間・空間スケールに分解して調査する必要がある。また、分野横断研究は膨大なデータから異なる物理量（太陽光量やオゾン濃度、震源の大きさなど）の因果関係を調べるため、それらのデータを柔軟かつ効率よく取り扱えることが望ましい。しかしながら、地球惑星科学は高まる専門性とともに分野が細分化され、各分野が独自に進歩を遂げてきたために、複数の分野のデータを確認することさえ容易ではない。例えば、観測データは測定を行った各機関で管理されていることが多く、異分野の研究者には、それらのデータの所在すら把握できないため、特定分野の研究だけの利用に留

まっている。また、主流なデータ形式は分野ごとに異なり、現象によってデータの可視化方法も異なるため、複数の分野のデータを確認するには、各分野の専用の読み出しツールやビューワなどを用意する必要がある。したがって、地球惑星科学分野におけるデータ利用の促進、ひいては分野横断研究を推進させるためには、次に示す研究インフラの整備を行い、データを素早く確認するためのシステム（本稿ではこれをクイックルックと呼ぶ）を構築し、広く一般に提供する必要がある。

【地球惑星科学分野におけるデータ利用を促進するための課題】

- ① 点在するデータの（所在の）一元管理
- ② 多様なデータ形式への対応
- ③ 様々な現象を柔軟に表現するビューワの提供

この問題を解決するために、各機関が蓄積した観測データのパラメータ（観測位置や時刻、測器の種類、データ形式などのメタ情報）を、インターネットを介して共有するシステム（Inter-university Upper atmosphere Global Observation NETwork; IUGONET⁷⁾）が構築されている。しかし、その研究領域は主に超高層大気限定されているため、更なる領域の拡張が必要である。また、IUGONETは多機能な統合解析ソフト⁸⁾も開発されているが、その機能を十分に利用するためには、商用のデータ解析用プログラミング言語（Interactive Data Language; IDL）を利用する必要があり、異分野の研究者や学生にとっては手を出し難いのが実情である。

そこで我々は、地球惑星科学分野の全体を対象にしたライセンスフリーのクイックルックサービスC3（Cross-Cutting Comparisons）の開発に着手した。C3は段階的な開発を進めており、既にサービスの一部を宇宙科学研究所のデータアーカイブ（Data ARchives and Transmission System; DARTS^{9,10,11)}）から一般に公開している。本稿では、上記の研究インフラの課題①、②に対処するためのシステムの概要を第2章に、第3章に課題③を解決するために工夫したサービスの特徴と今後の課題について記述する。

2 システムの概要

2.1 システム構成

C3 はデータサーバとウェブサーバで構成している (図 2)。データサーバには各分野の標準であり、かつ公開されているデータから定期的に格納している。一方、ウェブサーバは複数のマシンがあり、ウェブ関連のプログラムや文書をリポジトリ管理している。また、各マシンはデータサーバにある同じデータをマウントしている。

C3 の開発者がコミットした内容は、DARTS 運用者を通じて開発マシンから試験マシンに反映し、試験マシンで動作確認を行った後に公開している。公開マシン以外はインターネットから切り離しており、厳重に管理している。

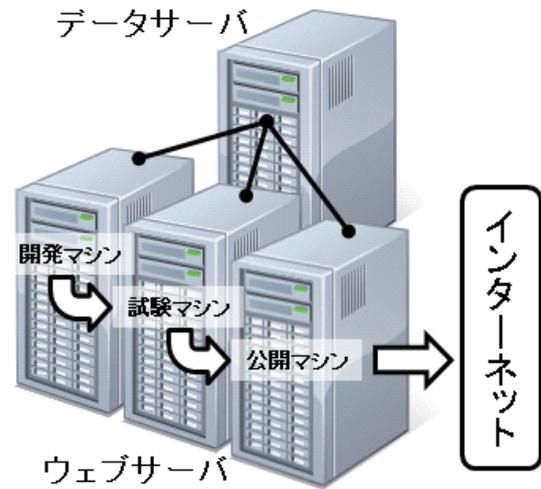


図2 システムの概要図

2.2 独自のデータファイルの管理

地球惑星科学分野のデータは1日単位でまとめられている場合が多いという点に着目し、C3では1日単位のデータファイルを取り扱っている。データ形式はアスキーとし、読み出し処理の短縮を図っている。また、そのファイルは図3に示すツリー構造で管理しており、データのパラメータを選択するインターフェースと一対一対応させている(3.2節を参照)。

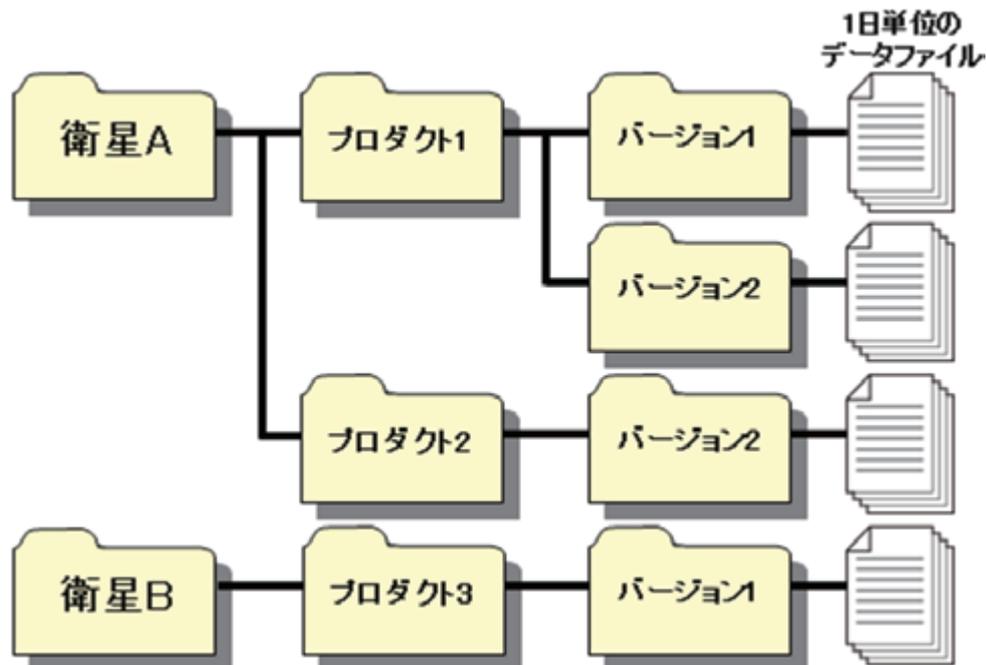


図3 データファイルの管理構成

2.3 通信処理の流れ

C3はユーザからの多量のリクエストを処理するために、ユーザが入力したリクエストの確認とデータの可視化をクライアント側で行っている(図4)。一方、リクエストに対するデータ処理・解析はウェブサーバで行い、その結果をクライアントに送信している。これによって、クライアントとサーバ間の通信は、1つのリクエストに対して1回のレスポンスで済み、ネットワークの通信速度による遅延を避け、サーバの負荷も低減させている。

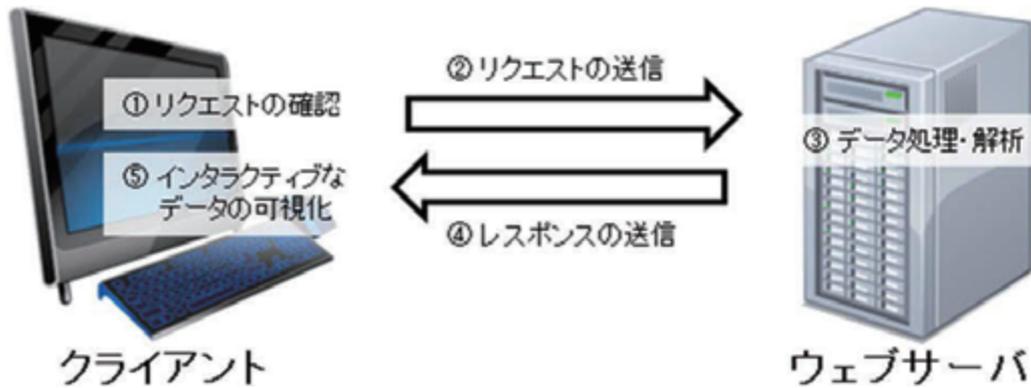


図4 通信処理の流れ

図中の通し番号は、ユーザがデータを選択し、可視化するまでの一連の順序である。

3 サービスの特徴と課題

C3 は既に地球惑星科学における 4 つの分野のデータを取り扱っており (表 1), その他の分野 (例えば超高層大気科学や大気電気学) のデータも順次追加する予定である. この章では, それらの多分野の膨大な情報量を分かり易く提供するためにした工夫を 3.1, 3.2 節に, 多様なデータを如何に柔軟に表現しているかを 3.3 節に, そして C3 を利用することで得られる情報の新たな管理と共有手段を 3.4 節に記述し, サービスを拡張するための今後の課題については 3.5 節にまとめている.

表 1 C3 が取り扱うデータセット (2015 年 7 月時点)

分野	プロダクト	データセット名
太陽物理学	全太陽放射照度, マグネシウム II	SORCE ¹²⁾
月惑星科学	日食 (最大食の様子, 影の様子)	Fred Espenak ¹³⁾
大気科学	大気温度, 大気微量成分	ACE-FTS ¹⁴⁾ , SMILES ¹⁵⁾
地震科学	マグニチュード, 震源の深さ	USGS ¹⁶⁾

3.1 ポップアップの説明表示

提供するサービスの機能が十分に利活用されるためには、それ相応の説明が必要である。しかしながら、多分野の情報を扱うサービスの場合は、専門用語が多いため、マニュアルの該当箇所を探しながら使用するには不便であり、良いクイックルックサービスとは言えない。そこで C3 はポップアップ機能を導入している。C3 のユーザインタフェースには、随所にインフォメーションマークのアイコンを設置しており、マウスカーソルを重ねる（オンマウス）によってその個所の説明を画面表示させている（図 5）。これは多くのウェブサービスでも活用されている機能であり、多分野を扱うウェブサービスにとっても、ユーザインタフェースをシンプルに保ちつつ、ユーザの必要な情報だけを瞬時に提供する有効な手法である。

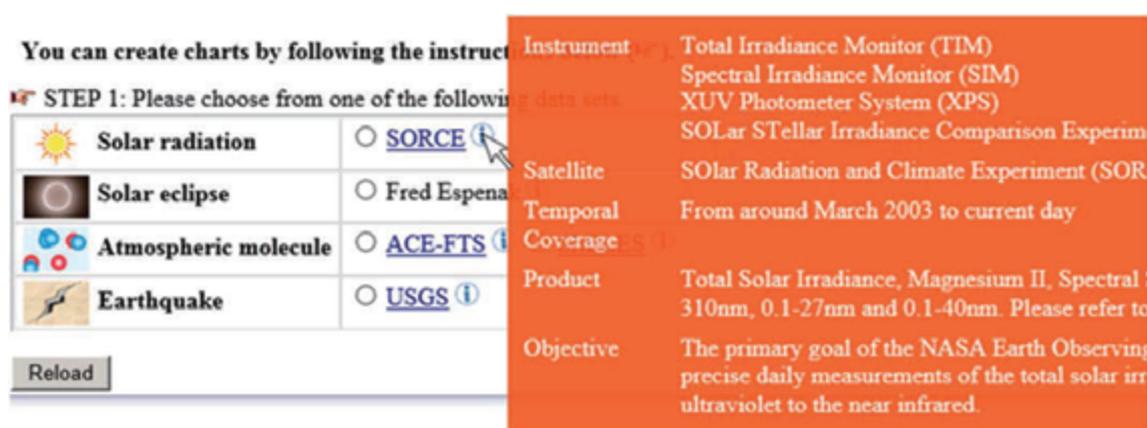


図 5 ポップアップの説明表示

インフォメーションマークにオンマウスして、説明文（図中の橙枠）をポップアップで表示させている様子。

3.2 誘導的な入力フォーム

一般的なクイックルックサービス^{17,18)}は、ブラウザにデータを可視化するために、まずそのデータを選択/入力し、次にそのデータに付随するパラメータを選択/入力する必要がある。したがって、利用可能なデータや図のタイプを豊富にすればするほど、パラメータの指定項目は増えるため、入力フォームは煩雑となってしまふ。これは多様なデータを取り扱うウェブサービスにとっては致命的であり、如何にシンプルな入力フォームを設計するかが大きな課題である。

しかしながら、パラメータを減らしてシンプルなインターフェースにした場合、簡易な操作は実現できるが、使用できるデータや機能を制限しなければならない。また、シンプルなインターフェースとして WolframAlpha¹⁹⁾が挙げられるが、シンプル過ぎるインターフェースでは、ユーザにとってどのようなサービスが利用できるのか分かり難い。

このように、ウェブサービスは“豊富な機能”と“簡易な操作性”のジレンマに陥りがちである。そこで我々は、データファイルがツリー構造で管理できることに着目し（2.1 節を参照）、パラメータの選択/入力も同じ階層構造とし、動的 HTML（サーバから HTML を改めてロードすることなく、ユーザの要求に応じて動的に HTML を書き換える）技術を利用した誘導的な操作性を実装している。

図 6 (a) は C3 が提供するインターフェース、図 6 (b) はパラメータの階層遷移を示している。初

期画面はデータセットの選択であり、ユーザの選択に従って、次の入力フォーム（STEP 2）が表示され、そこで入力された内容に従って、その次の入力フォーム（STEP 3）が表示される。このように誘導的な操作性によって、ユーザの入力を支援するだけでなく、画面には必要な情報しか表示されないため、シンプルなインターフェースも提供している。また、データファイルが同じ構造で管理されているため、ユーザはデータファイルを直接選択することになり、データ検索の高速化も実現している。

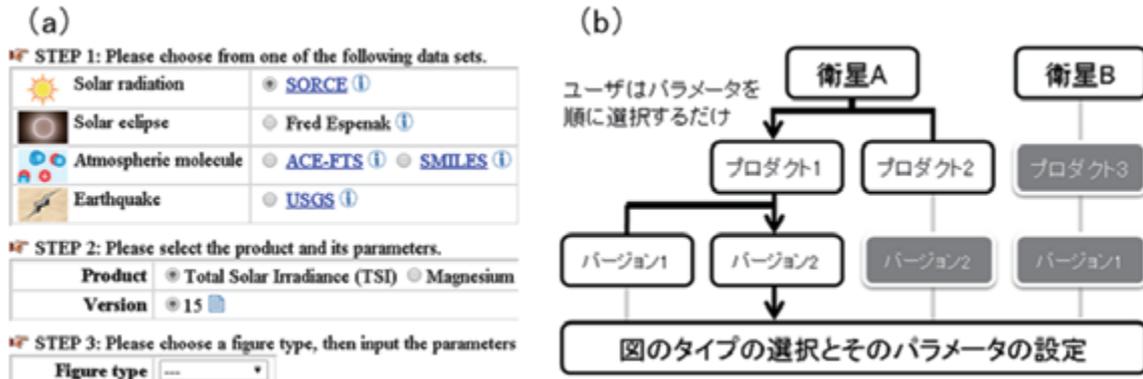


図 6 (a) データとパラメータの選択画面と (b) その階層遷移を示した簡易設計

(b)の矢印はユーザの選択フローを示している。また、黒文字は画面に表示されるパラメータ、白文字はユーザが選択しなければ表示されないパラメータである。

3.3 多様なデータを柔軟に可視化

1章に記述したように、太陽地球環境内で起きる突発現象や長期変動が他にもたらす影響を捉えるためには、その物理・化学素過程を様々な時間・空間スケールに表現するビューが必要である。そのため、C3ではHTMLとJavaScriptベースのライブラリの一つ、Highcharts²⁰を利用してデータを可視化している。図7の(a)から(d)はC3が提供する図の例である。図に示すように、点、直線、面、そして立体の表現ができるため、ほとんどの科学データの可視化が可能である。また、ユーザが作成した図は様々なファイル形式（PNGやJPEG、PDF、SVG）で出力することもできる。

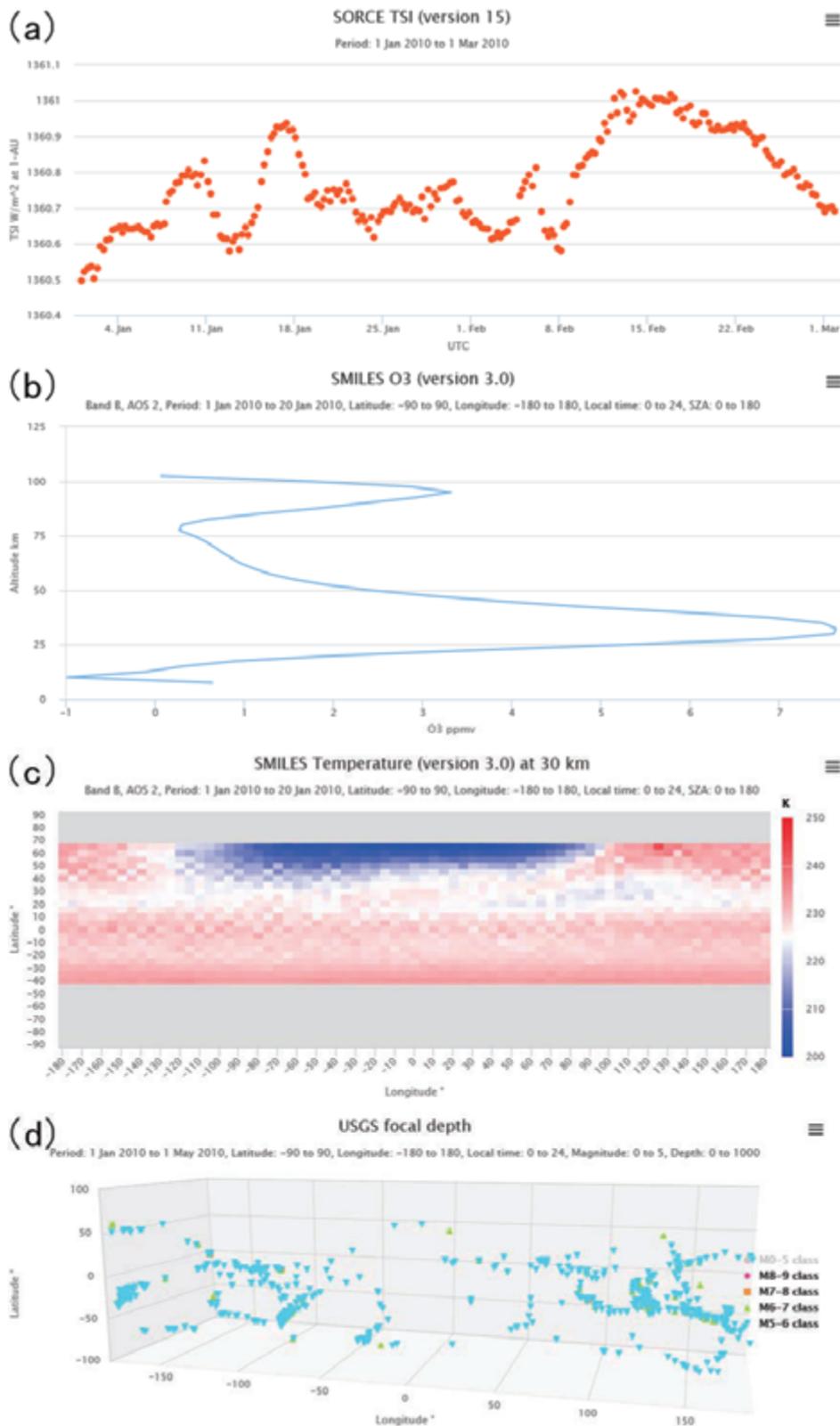


図7 C3が提供する図の例1：(a) 太陽活動の様子，(b) オゾンの高度分布，(c) 温度の緯度経度断面図，(d) 震源の深さ

次に C3 を活用した分野横断研究の一例として、太陽放射量の変化がもたらす大気オゾンへの影響を、日食を利用して調査した内容²¹⁾を紹介する。図 8 (a) は 2010 年 1 月 15 日の日食の時に地球に投影された月の影の様子を示しており、図 8 (b) は、その時の大気科学センサ SMILES の観測点を示している。そして、図 8 (a) と (b) を比較することで、日食時に月の影となった領域を SMILES が観測していたことが分かる。このように複数の図 (ウィンドウ) を並べることによって、異なる (ここでは月惑星科学と大気科学の) データを比較することができる。

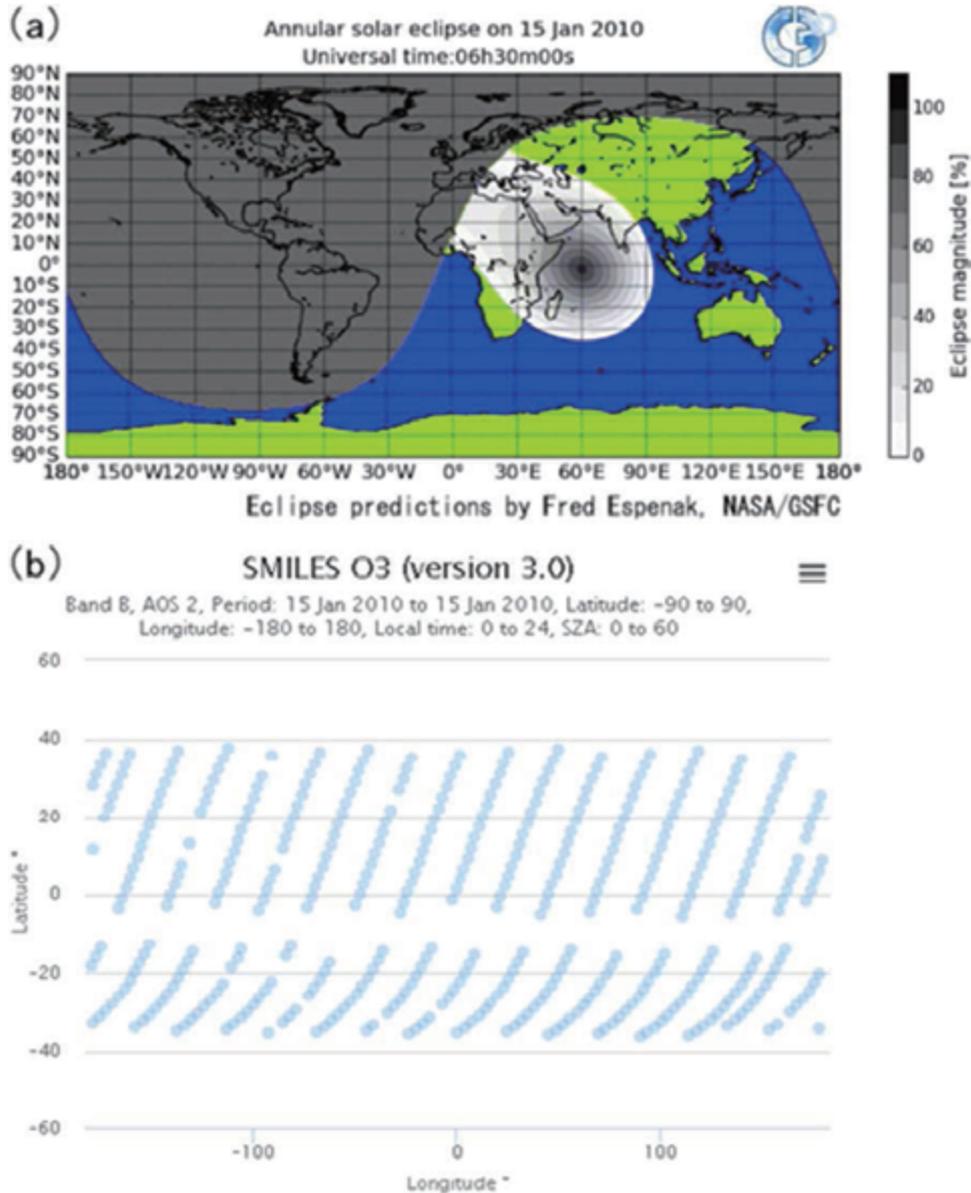


図 8 C3 が提供する図の例 2 : (a) 日食時の月の影の様子, (b) SMILES の観測点
(a) のグレースケールは食分率を表している. (b) 青丸印は SMILES の観測点である.

続いて図9は図8に示したSMILESの観測データで、日食時の減光に伴い高度60 kmのオゾン混合比が増加した様子を示している。図に示すように、より詳細に見たい範囲をドラッグすることで、その領域がズームアップされ、さらにデータ点にオンマウスすることで、その他の情報（例えば観測日時や観測値など）も知ることができる。このようなインタラクティブな操作性もC3が提供するビューワの特徴であり、様々な現象を柔軟に表現している。

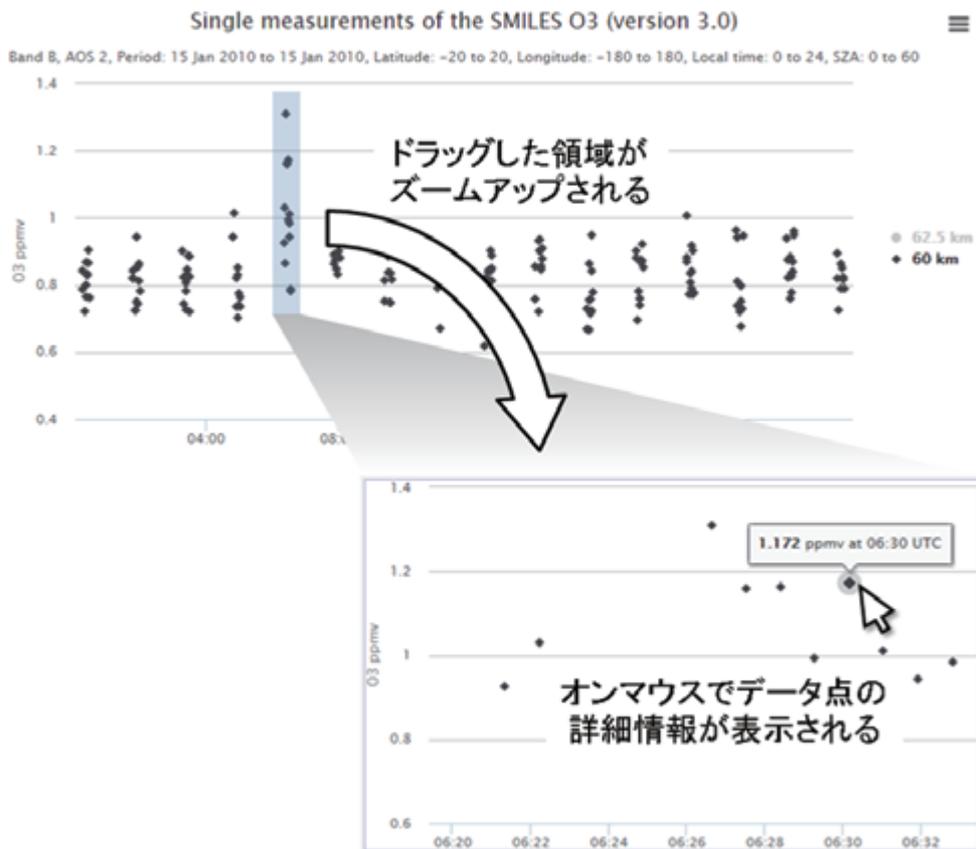


図9 C3が提供する図の例3：日食時のオゾン混合比を調べている様子

3.4 クエリ文字列の活用

広く自然科学の研究では、ある現象の理解を深めるために図を利用するが、分野横断研究では、より多量の図を必要とするため、それらの図の管理が難しくなることがある（図 10a）。また、それらの資料を共同研究者と共有するのはなおのこと困難である。そこで、この問題を解決するために、C3で作成される全ての図には、ブラウザのアドレスバーに URL のクエリ文字列を付けている。そのクエリは CGI プログラムの命令文であり、同じクエリを用いれば、いつでもブラウザに同じ図を再現することができる。また、クエリは単なる文字列であるため、ハイパーテキストのようにリンクとして文書ファイルにまとめれば（図 10b）、図の内容をファイル名だけで識別する必要がなくなり、図の保存がなくなる分だけデータ容量も抑えられる。さらに、クエリは Twitter などのテキストベースの通信サービスを利用して発信できるため、共同研究者との情報共有の新たな手段としても有効である（図 11）。

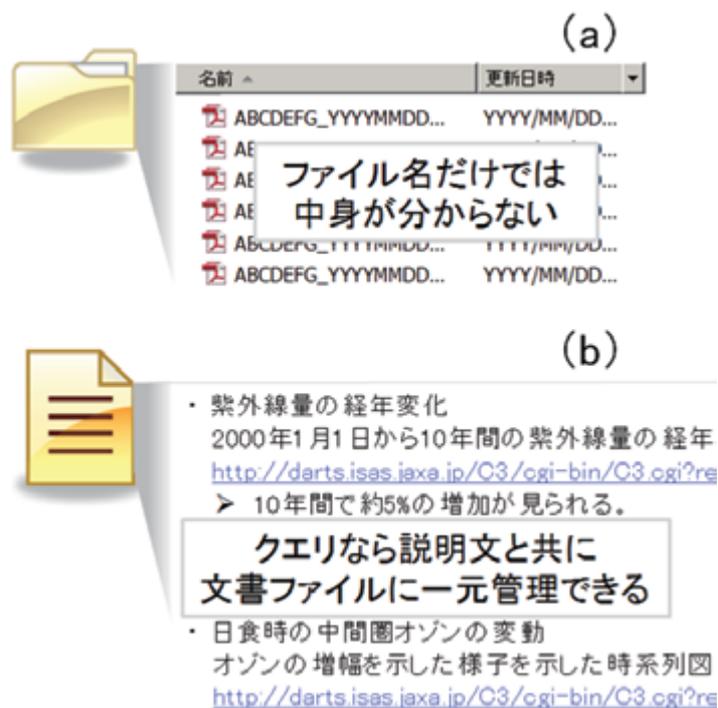


図 10 (a) 画像ファイルの一覧と (b) クエリの管理例

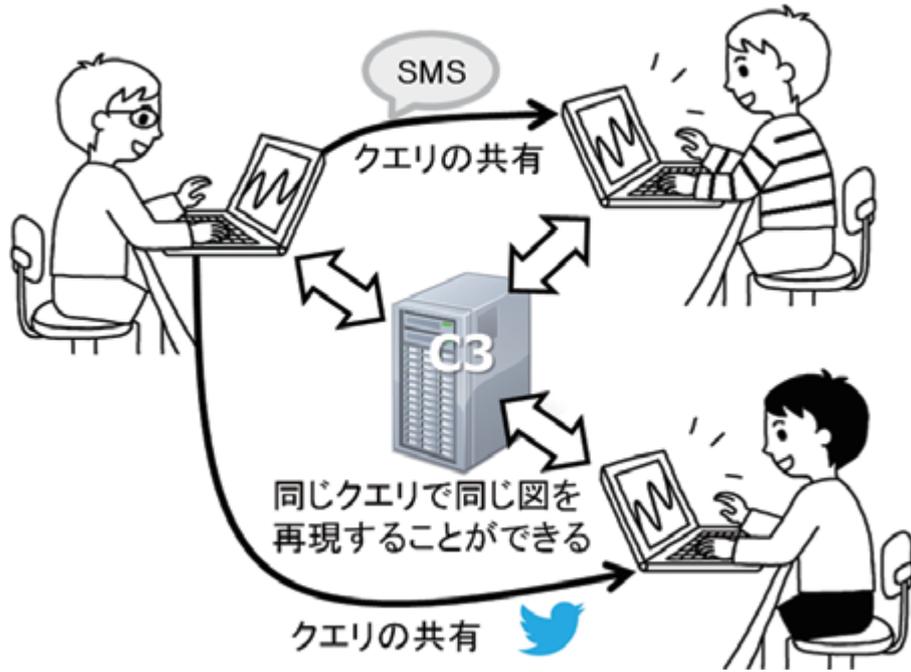


図 11 クエリの共有による同じ図の再現

3.5 サービスの課題と解決策

3.5.1 より直観的なインターフェース

現在、C3は地球惑星科学における4つの分野のデータを取り扱っているが、その数が増えることを想定してインターフェースにも工夫を凝らしている。例えば、各分野にはイラストを付けており、その順番を地球の地表面から高い順（つまり上から太陽、大気、地震）としている。また、インターフェースにはデータセット名を明記しているため、そのデータを頻繁に使うユーザにとっては効率的である（図5、図6(a)を参照）。しかしながら、それらのデータセットに馴染みのないユーザにとっては、数が増えるにつれインターフェースが煩雑に感じると想像される。

そこで、より直観的なインターフェースとして、データセットを画像から選択する機能を追加することを予定している。図12はそのデザイン案を示している。画像に記載されている現象名や物質名にオンマウスすることで、そのデータを取り扱うデータセット名がポップアップで表示されるため、データセットを知らなくてもデータの確認ができる仕組みである。

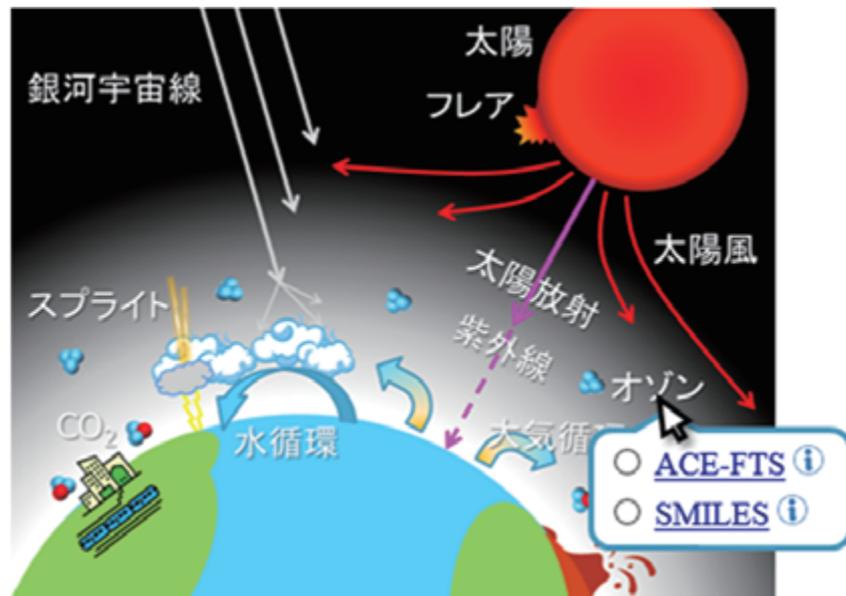


図12 より直観的なインターフェース

画像中の物質名（オゾン）にオンマウスして、データセット（ACE-FTSとSMILES）を表示させている様子。

3.5.2 非公開データの取り扱い

C3には地球惑星科学における各分野で標準的なデータから追加しているが、そのデータはユーザの使用用途を制限しないために、公開されているものに限定している。また、2.1節に記載したように、セキュリティの問題上、データサーバはインターネットから切り離しているため、ユーザが個人的に持つような公開されていないデータに関しては、C3を用いて可視化することはできない。この問題を解決するために、ローカルファイルを（データサーバにアップロードするのではなく）、C3に（CSVやJSONなどのアスキー）ファイルを読み込む機能を実装し、ユーザのブラウザで取り扱えるようにすることを検討している。これによって、C3が提供するデータとの比較やビューワの利用が実現される。

3.5.3 共同の研究ノート

大気化学にみられる連鎖反応のように、太陽地球環境内における多くの現象は、大気を構成する様々な要素に伝搬し、巡り巡ってそのもの自身にも影響を与えるため、ある特定の分野だけでは研究を完結することはできず、分野間の連携が極めて重要である。そのためC3は、コラボレーションツールとして、分野間の隔たりのない共用のウェブページを開設する予定である。これは3.4節に記述したハイパーテキストの拡張であり、Wikiのように自由に文章の閲覧や編集ができるだけでなく、ページ上のクエリを変更することによって、図の編集も行える。まさに“共同の研究ノート”である。これによって、分野間の知識が共有され、研究がよりスムーズに推進されると考える。ただし、このようなユーザ参加型のサービスにはセキュリティ上の問題があるため、アカウントの認証、もしくはIPアドレスと変更箇所を記録する必要があるため、セキュリティと利便性のトレードオフが課題である。

4 まとめ

我々は地球惑星科学分野におけるデータ利用を促進すべく、ウェブサービス C3 の開発を進めている。そのインタラクティブな操作性によってユーザの入力を支援し、多様なデータを柔軟に可視化するビューワを提供している。C3 はライセンスフリーであるため、一般ユーザの即時利用が可能である。(まだ開発途上ではあるが) その便利さ故に、研究者だけでなく、教育機関や一般にも利用され、公開して約 1 年間でユーザ数は 600 名を突破している (2015 年 7 月時点)。C3 によって太陽地球環境の総合的な理解が大きく前進することを期待している。

謝辞

C3 は宇宙航空研究開発機構・宇宙科学研究所、科学衛星運用・データ利用センター (C-SODA)、京都大学生存圏研究所 (RISH)・生存圏科学萌芽研究からの助成を受けて実施しています。ここに感謝の意を表します。

参考文献

- 1) Shindell, D., D. Rind, N. Balachandran, J. Lean, and P. Lonergan, Solar cycle variability, ozone, and climate. *Science*, 284, 305-308, 1999, doi:10.1126/science.284.5412.305.
- 2) Meehl, G. A., J. M. Arblaster, K. Matthes, F. Sassi, and H. van Loon, Amplifying the Pacific Climate System Response to a Small 11-Year Solar Cycle Forcing, *Science* 325, 5944, 1114, 2009, doi:10.1126/science.1172872.
- 3) Ouzounov, D., S. Pulinet, A. Romanov, A. Romanov, K. Tsybulya, D. Davidenko, M. Kafatos, P. Taylor, Atmosphere-ionosphere response to the M9 Tohoku earthquake revealed by multi-instrument space-borne and ground observations: Preliminary results, *Earthquake Science*, 24, 6, 557-564, 2011, doi:10.1007/s11589-011-0817-z.
- 4) Arnone, E., A. K. Smith, C.-F. Enell, A. Kero, and B. M. Dinelli, WACCM climate chemistry sensitivity to sprite perturbations, *J. Geophys. Res. Atmos.*, 119, 6958-6970, 2014, doi:10.1002/2013JD020825.
- 5) Prenni, A. J., P. J. DeMott, A. P. Sullivan, R. C. Sullivan, S. M. Kreidenweis, and D. C. Rogers, Biomass burning as a potential source for atmospheric ice nuclei: Western wildfires and prescribed burns, *Geophys. Res. Lett.*, 39, L11805, 2012, doi:10.1029/2012GL051915.
- 6) Petersen, G. N., H. Bjornsson, and P. Arason, The impact of the atmosphere on the Eyjafjallajökull 2010 eruption plume, *J. Geophys. Res.*, 2012, 117, D00U07, doi:10.1029/2011JD016762.
- 7) Yatagai, A., Y. Tanaka, S. Abe, A. Shinbori, M. Yagi, S. UeNo, Y. Koyama, N. Umemura, M. Nose, T. Hori, Y. Sato, N. O. Hashiguchi, N. Kaneda and IUGONET project team, Interuniversity upper atmosphere global observation network (IUGONET) metadata database and analysis software, *Data Sci. J.*, 13, PDA37-PDA43, 2014.
- 8) 田中良昌, 新堀淳樹, 鍵谷将人, 堀智昭, 阿部修司, 小山幸伸, 林寛生, 吉田大紀, 河野貴久, 上野悟, 金田直樹, 米田瑞生, 田所裕康, 元場哲郎, 三好由純, 関華奈子, 宮下幸長, 瀬川朋紀, 小川泰信, IUGONET 解析ソフトウェアの開発, *宇宙科学情報解析論文誌*, 1, 91-98, 2012.
- 9) Miura, A., I. Shinohara, K. Matsuzaki, F. Nagase, H. Negoro, S. Uno, et al., In: Manset, N., Veillet, C., Crabtree, D. (eds) *Astronomical Data Analysis Software and Systems (ADASS) IX*, ASP Conf. Ser.,

- 216, 180, 2000.
- 10) Tamura, T., H. Baba, K. Matsuzaki; A. Miura; I. Shinohara, F. Nagase, M. Fukushi; K. Uchida, Data Archive and Transfer System (DARTS) of ISAS, ASP Conf. Ser. 314, ADASS XIII, ed. F. Ochsenbein, M. Allen & D. Egret (San Francisco: ASP), 22, 2004.
 - 11) <http://www.darts.isas.jaxa.jp/>
 - 12) Rottman, G., The SORCE Mission, *Solar Phys.*, 230, 1, 7-25, 2005.
 - 13) <http://www.mreclipse.com/>
 - 14) Bernath, P., et al., Atmospheric Chemistry Experiment (ACE): Mission overview, *Geophys. Res. Lett.* 32, L15S01, 2005.
 - 15) Kikuchi, K. et al., Overview and Early Results of the Superconducting Submillimeter-Wave Limb-Emission Sounder (SMILES), *J. Geophys. Res.*, 115, D23306, 2010, doi:10.1029/2010JD014379.
 - 16) <http://www.usgs.gov/>
 - 17) <http://lasp.colorado.edu/home/sorce/data/>
 - 18) <http://mls.jpl.nasa.gov/>
 - 19) <http://www.wolframalpha.com/>
 - 20) <http://www.highcharts.com/>
 - 21) Imai, K., T. Imamura, K. Takahashi, H. Akiyoshi, Y. Yamashita, M. Suzuki, K. Ebisawa and M. Shiotani, SMILES observations of mesospheric ozone during the solar eclipse, *Geophys. Res. Lett.*, 42, 2015, doi:10.1002/2015GL063323.

適切かつ高度なデータ処理環境を提供するための ウェブインターフェース “Web Processing Catalog” かぐや搭載スペクトルプロファイラデータを事例として

林 洋平^{*1} 山本幸生^{*1} 小川 佳子^{*2} 平田 成^{*2} 出村 裕英^{*2} 寺菌 淳也^{*2} 松永 恒雄^{*3}

“Web Processing Catalog” : Web interface for distribution of
the appropriate and advanced environment for the data processing

– A case study for the data from the Spectral Profiler on board Japanese satellite Kaguya

HAYASHI Yohei, YAMAMOTO Yukio^{*1}, OGAWA Yoshiko^{*2}, HIRATA Naru^{*2},
DEMURA Hirohide^{*2}, TERAZONO Junya^{*2}, and MATSUNAGA Tsuneo^{*3}

Abstract:

In the planetary science, many observation data are available via internet. To get some information from the observation data, a lot of steps of data processing are necessary. Either the data providers or users have to handle such processes. For the case where the providers are responsible, the data processing is a black box for the users usually, so the potential of the data could not be fully utilized. On the other hand, for the case the users themselves follow the processes, the data usability would be the problem, because the users need to know many technical things and they sometimes could not afford that. We focus on the importance of the relationship between the data and data processing. We propose a new protocol “Web Processing Catalog” for developing web interface to distributing the metadata and desirable programs for data processing. Finally, we also implement a preliminary system and conduct verification where we use the actual data from the spectral profiler onboard Japanese lunar orbiter, Kaguya.

keywords: Data Processing, Scientific Data, Metadata, Web Service, Web Processing Catalog, GEKKO, Kaguya (SELENE), Spectral Profiler

概要

惑星科学の分野ではウェブ上のサービスを通じて科学データが利用可能である。科学データから情報を得るためには目的に応じてデータ処理がなされる必要がある。このデータ処理はサービス提供者あるいはユーザ自らが行う必要がある。前者の場合、データ処理のプロセスはブラックボックスになりがちであり、データが本来もつ可能性を發揮しきれない可能性がある。後者では、データの利用には多くの専門知識を必要とする場合があり利用が阻まれる場合もある。そこで上記の問題を解決するため、データとデータ処理プログラムを関連付ける重要性を明らかにする。そして、ユーザへ適切かつ高度なデータ処理環境を提供することを目指し、両者を関連付けるメタデータおよびデータ処理プログラムを配信するウェブインターフェースを構築するためのプロトコル “Web Processing Catalog” を提案する。また、実際の月探査衛星「かぐや」に搭載されたスペクトルプロファイラ (SP) のデータを用いた試験的なシステムの実装を行い検証をおこなう。

1. はじめに

惑星科学の研究対象である太陽系の天体は望遠鏡や探査衛星によって様々な観測がなされ、多くの科学データが蓄積されている。本論における科学データは、可視化処理がなされていないデータを指し、様々なデータ処理により様々な情報が取

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

^{*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)

^{*2} 会津大学先端情報科学研究センター

(Research Center for Advanced Information Science and Technology, The University of Aizu)

^{*3} 国立環境研究所 (National Institute for Environmental Studies)

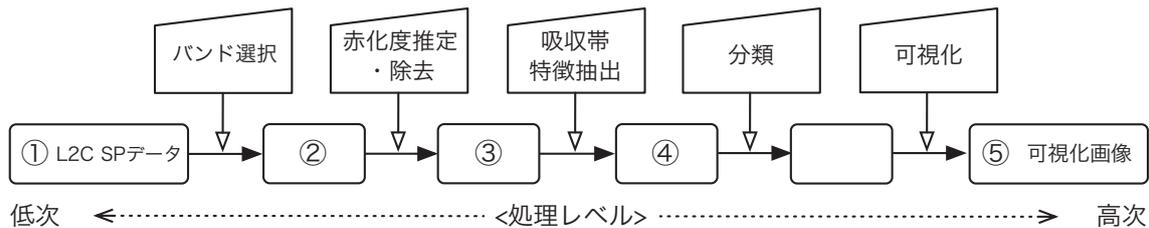


図 1 データ利用に至るまでのデータ処理プロセス

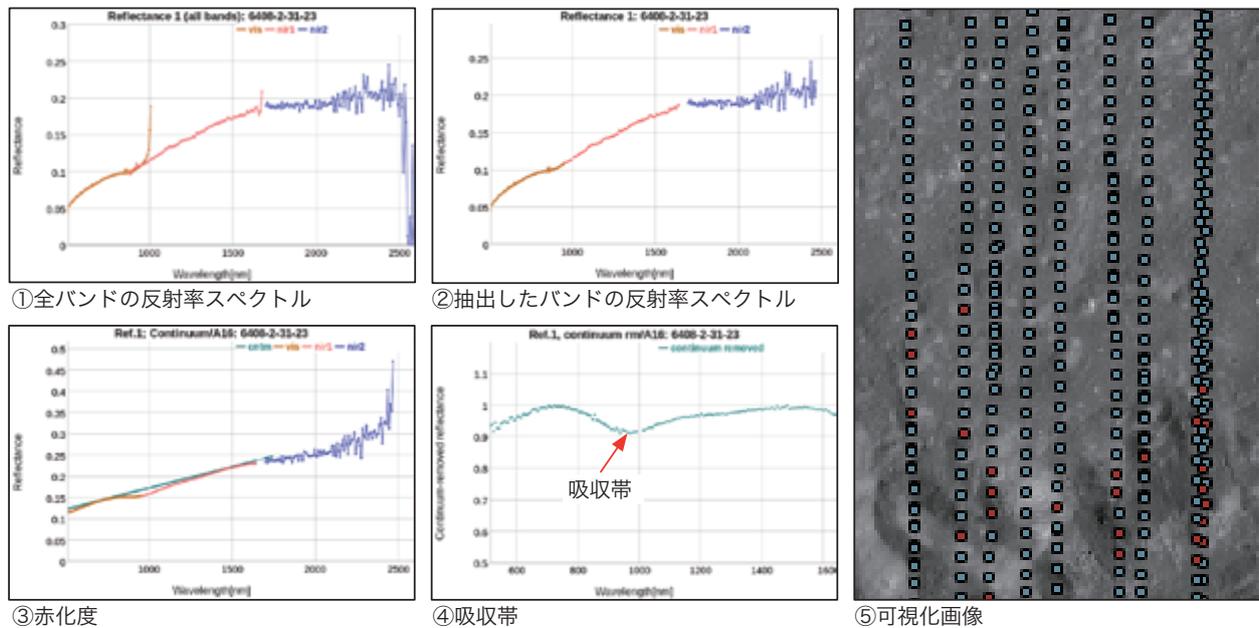


図 2 図 1①～⑤のデータについてそれぞれ可視化したグラフおよび画像

り出せる可能性を持っているものとする。科学データが得られている天体は、大気が薄い火星や地球の月のように変化が比較的少ない天体がある一方で、金星や木星のように濃い大気に覆われ刻一刻と変化する天体があり、イトカワのように球体ではない比較的小さな天体もある。大小・性質も様々な多くの惑星が観測の対象となっており、それぞれ観測対象特有の現象を明らかにするためには、様々な観測機器によってもたらされる科学データを活用する必要がある。地球観測で用いられていない観測機器を用いて観測が行われることもある。一般的に利用されていない最先端のセンサーあるいは特殊なセンサーのデータは、データの取り扱いが可能な人材に限られると推測される。

例えば、かぐやデータアーカイブ¹⁾で配信されているプロダクトレベル 2 (L2C) の SP データ (図 1 ①) の場合、鉦物分布情報を読み出すためには図 1 に示したように「バンド選択」、「赤化度推定・除去」、「吸収帯特徴抽出」、「分類」、「可視化」といったデータ処理が必要となる。これらの一連のデータ処理を科学データに適用することによって、図 2⑤に示した画像が得られる。データ処理により変化する実際のデータについてわかりやすくするため、図 1①～⑤のデータについてそれぞれ可視化したグラフを図 2 へ参考として掲載した。これらのグラフは「月光」(後述)で作成した。図 2④の「吸収帯」は、「赤化度推定・除去」を行ったデータにおける窪みの最深値を示す波長であり、矢印で示した部分になる。図 2⑤の「可視化画像」は、分類のデータ処理の結果に基づきグループごとに色を割当て、別途用意した Lunar Reconnaissance Orbiter 搭載の Wide Angle Camera (WAC)²⁾の画像上へ四角い点としてプロットしたものになる。図 ⑤の四角い点のひとつひとつが SP の観測点であり、四角い点が同色であればスペクトルが似通っていることを示している。このようなデータ処理はデータへの理解がなければ難しく、ノウハウの固まりであるといえる。ユーザが科学データを利用するためには、使用目的に応じてデータ処理を行い、画像やグラフなど可視化された状態にする必要がある。

このような状況に対して林ら³⁾はウェブアプリケーション「月光」⁴⁾を開発した。「月光」は、SP データをウェブブラウザから閲覧できる地理情報システム (GIS) である。SP は可視から近赤外までの波長域 (0.5～2.6 μm) で月面からの反射光を 296 バンドで連続的にとらえる分解能を有している⁵⁾⁶⁾。「月光」では、全 SP データ (L2C) を搭載しており、ユーザはオリジナルの生データをはじめ、「利用に適したバンドの抽出」や「ノイズ除去」、「赤化度推定・除去」などのデータ処理を行なったグラフを閲覧できる。現在、著者らは「月光」上で SP データの本格解析を可能にするためのデータ処理機能の拡張を検討している⁷⁾。現在の「月光」では上記に示した若干のデータ処理機能を備えているが、「月光」上でユー

ザ自らが用意したデータ処理プログラムを適用することはできない。逆に「月光」のデータ処理プログラムを「月光」以外で用いることもできない。著者らは「月光」の拡張にあたり、「月光」で完結しないオープンなシステムづくりを目指している。そこで、ウェブサービスにおいて適切かつ高度なデータ処理を実現するため、既存のウェブサービスで用いられている地図データや科学データの配信に用いられている代表的なプロトコルを検証しつつ、新たなプロトコルの提案を行う。

2. 各種プロトコルにおけるユーザによるデータとデータ処理への関与

多くの科学データはインターネットによって配信されている。配信側とユーザ側はインターネットによって隔てられており、データ配信のためのプロトコルを用いて配信が行われる。また、科学データはユーザによって利用されるまでの間に必ず何かしらのデータ処理が行なわれている。データ処理は、配信側、ユーザ側、あるいはその両者が関わっている。データ処理がどこで、どの程度おこなわれるかによって、ユーザのデータ処理への関わりが大きく変わってくる。本章ではモデル図を示しつつ、地図配信が可能なプロトコルを3種類とりあげ、科学データからデータ処理を経て、可視化画像へ至るまでのユーザのデータとデータ処理への関与について述べる。

2.1 モデル図の凡例

図3以降のモデル図の解説を行う。各モデル図の左側の「サービス提供側」としているブロックはサービス提供者のコンピュータ環境であり、右側の「ユーザ側」のブロックはユーザのコンピュータ環境になる。コンピュータ環境はサービス提供側とユーザ側でそれぞれひとつにまとめられているが、実際は分散環境であったり、サービス提供側がデータ処理を行っていない場合もある。しかしながら本議論の本質ではないため図のようにサービス提供側とユーザ側とに単純化して取り扱うこととする。

「角が取れた四角」は「データ」を、「台形」は「プログラム」を表している。可視化されたデータを閲覧するためにはGISアプリケーションなどのアプリケーションが必要だが、コンピュータ環境においてデータ処理以外のこれらのプログラムについては省略する。

サービス提供側とユーザ側にまたがっている「薄く実線の矢印」は「インターネットでのデータ配信」を示す。水平に伸びている「黒く塗りつぶされた矢印」は、「矢印の始点につながっているデータから終点につながっているデータへの変換」を示している。垂直に伸びている「白抜きの矢印」は前記の変換に際して「データ処理の適用」を示している。各種「矢印」や「データ」「プログラム」について、実線の場合はユーザが操作できることを示し、点線の場合は、ユーザが操作できないことを表している。

2.2 Web Map Service

Web Map Service (WMS)⁸⁾ は、ユーザがインターネットを通じて地図画像を取得するインターフェースを定めた標準プロトコルである。地図画像は位置情報を持った画像のことである。地図画像は、本論が対象とする科学データから作成される可視化画像以外にも、ベクターデータによって作成される地質図や道路地図なども含む。アメリカ地質調査所⁹⁾ やNASAのジェット推進研究所 (JPL)¹⁰⁾ は、WMSを用いることにより月や火星を始め探査が行われた天体の衛星画像や地形図などを配信している。ユーザは、WMSにおいて定義されている「GetMap」リクエストを用いてデータリソース、座標系 (CRS) および空間的範囲を示したバウンディングボックス (BBOX) などの情報をサーバへ送ることによって、レスポンスとして地図画像を得ることができる。

ここから、データ配信にWMSが用いられている場合におけるデータ処理について検証する。図3(a)に示すように、サービス提供側のデータ処理にはユーザ側は関与できない。WMSを用いて可視化画像を行う場合、配信に先立ちサービス提供側でデータ処理が必要となる。データ処理が専門家の監修のもとで行われている場合、妥当性が保証されているといえる。

また、配信後のデータ処理に関して、WMSは基本的にはウェブブラウザで用いられることが想定されており、配信後の画像に対してデータ処理を行うことは想定されていない。WMS 1.3.0¹¹⁾ では、地図画像のフォーマットとして、JIF、PNG、JPEG、TIFFが示されている。多次元ラスタや科学データをそのまま画素値として割り当てた画像、特に符号なし整数型以外のデータ型の画像などは規格で想定している利用の範囲を越える。

ところでWMSには、「GetCapabilities」リクエストがあり、サービス提供者はメタデータを配信することができる。そこで「abstract」の項目を用い、データ処理に関することをユーザ側に伝えることは可能である。例えば、JPLのonMoon¹⁰⁾ は、「abstract」を通じてデータ処理に関する情報を自由記述文で配信している。

2.3 Web Coverage Service

Web Coverage Service (WCS)¹²⁾ は、ユーザがインターネットを通じて地理空間におけるカバレッジ (被覆) データを取得するインターフェースを定めた標準プロトコルである。WCSを用いて配信されるカバレッジデータは、ユーザ側でデータ処理を行うことができる可視化処理が行われていないデータである。ユーザは、WCSにおいて定義されている

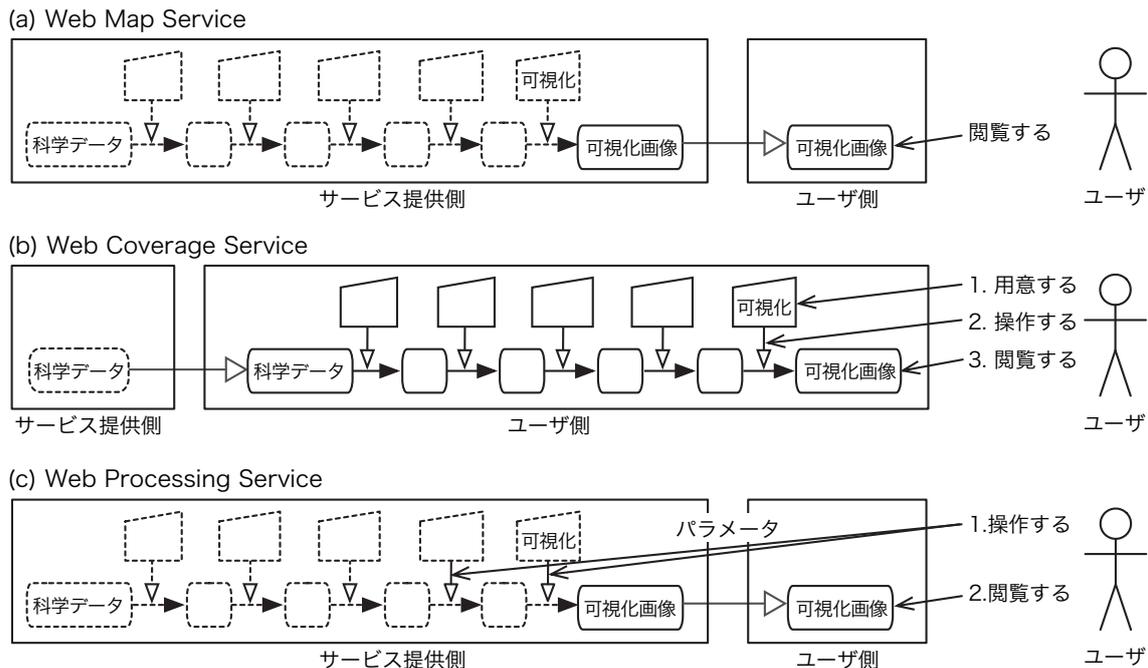


図 3 各プロトコルにおけるユーザとデータ処理の関係を示した模式図

「GetCoverage」リクエストを用いて CRS や BBOX などの情報に加え、それぞれのデータ特有の時間軸や特定の波長といった詳細な切り出しの情報などをサーバへ送ることによって、レスポンスとしてカバレッジデータを得ることができる。

ここから、データ配信に WCS が用いられている場合におけるデータ処理について検証する。WCS では配信用のデータフォーマットとして GeoTIFF, NetCDF, JPEG2000 などを明示しており、これらは科学データの格納に適している。図 3(b) に示すように、WCS で配信されるデータは可視化処理が行われていないため、ユーザがデータ処理に直接的に関与することになる。すなわち、ユーザは多くのデータ処理についてプログラミングをするか既存のプログラムを用意する必要があり、データに関する様々なノウハウがユーザには要求される。その一方で、様々なデータ処理を駆使することにより様々な可視化が可能であり、データのもつ可能性を十分に引き出せる。そのためデータ操作の自由度が高いといえる。その半面、ユーザ側のデータ処理の妥当性については保証されない。

2.4 Web Processing Service

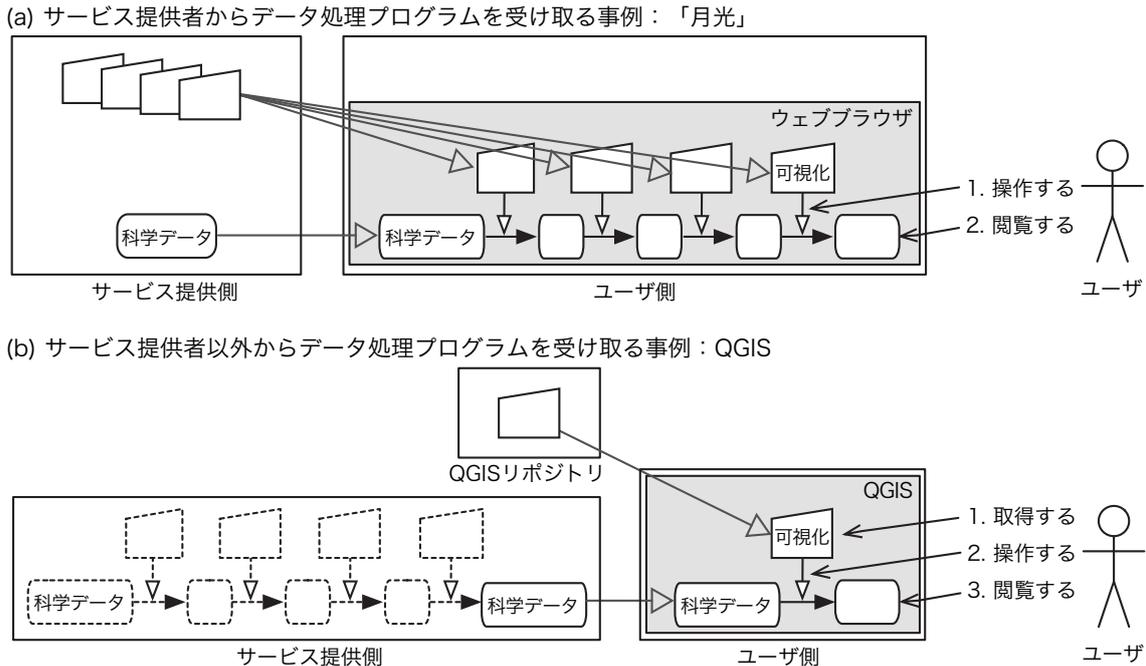
Web Processing Service (WPS)¹³⁾ は、サーバーのデータ処理をインターネットを通じて実施するインターフェースを定めた標準プロトコルである。ユーザは、WPS において定義されている「DescribeProcess」リクエストの実行によってデータ処理に必要なパラメータ値の情報を得られる。「Execute」リクエストを用いてパラメータを指定することによってデータ処理を実行できる。「GetResult」リクエストによってデータ処理結果を得ることができる。

ここから、データ配信に WPS が用いられている場合におけるデータ処理について検証する。図 3(c) に示すように、ユーザは可視化された地図画像をインターネットを通じて得られるという点では WMS と同様だが、上述のようにデータ処理の部分についてパラメータをサーバ側に渡すことによって関与できるという点が大きく異なる。だが、WPS を通じてサーバで実行することが可能なデータ処理は、あらかじめ配信側が用意した特定のデータ処理に限られる。図 3(c) の例では、配信側に 5 つのデータ処理があるが、そのうち右側 2 つのデータ処理にユーザが関与していることを表現している。その他のデータ処理にはユーザが関与できないことを示している。

WPS では、ユーザがデータに関するノウハウを持ち合わせていなくとも、ある程度データ処理に関与できる。図 3(b) の WCS のようにデータ処理の変更を自由にできるというわけではない。また図 3(a) の WMS のように、科学データから抽出可能な情報の自由度が一切ないという状態でもない。WPS では、制限付きではあるが動的にデータ処理が反映された可視化画像を得ることができる。

どのようなデータ処理が行われるかは、パラメータが設定できる部分に関してはおおよその推測が立つので、データ処理の明示性はある程度は保たれる。とはいえ配信側のデータ処理をユーザ側で完全に再現することは困難である。またパラメータ以外の部分(図 3(c) の点線部分)はサービス提供者が明示していない限りは不明である。

なおデータ配信側でのデータ処理は、データ特性を知り尽くした専門家の監修のもと作成されており、妥当性が保証されているといえる。



2.5 既存のプロトコルの課題

本章では3つのプロトコルごとに、ユーザの側からみたデータ処理に注目した。データ処理が配信側かユーザ側のどちらで行われるかによって、データの持つポテンシャルと学習コストなどのユーザ負荷が変わり、両者はトレードオフの関係であるといえる。科学データの持つ可能性、すなわち情報を引き出すためにはユーザがデータ処理に強く関与する必要がある。WMSでは、データへの関与はできず、科学データのもつ一側面の情報のみ得られるが、ユーザのデータ処理への負荷が全くない。WCSでは、データへの関与ができ、科学データの多面的な情報を引き出せる可能性があるものの、ユーザのデータ処理への負担が高い。WPSでは、ユーザの負荷を抑えつつデータ処理への関与が可能だが、科学データから引き出せる情報は限定的である。

3. データ処理プログラムの配信事例

科学データの持つポテンシャルを引き出すためには、ユーザのデータに対する関与が不可欠である。一方で、科学データに適したデータ処理プログラムを準備し適用することについては、ユーザ側への負荷が高い。そこで、科学データに適した処理プログラムを配信すればユーザ側への負荷を下げつつ、データの持つポテンシャルを引き出せる可能性がでてくる。そこで、ユーザ側へデータ処理プログラムを配信している事例をみていくことにする。なお、ウェブGISにおいてデータ処理プログラムを配信するインターフェースを定めたプロトコルは現状では存在しない。配信側のデータ処理のプログラムをユーザに配信し、ユーザ側でデータ処理を反映できればデータの持つポテンシャルを保ちつつ、ユーザ負荷を下げるができる。その実現のために本章では利用可能な既存のデータ処理のプログラムについて検証していく。

3.1 サービス提供者からデータ処理プログラムを受け取る事例：「月光」

先述の「月光」はデータ処理プログラムを受け取れる。月光の起動とともに、データ処理プログラムはサーバからダウンロードされ、ユーザの指示によってデータ処理プログラムが科学データに適用され、グラフという形でデータの可視化が行われる。「月光」では、図4(a)のように、データ処理プログラムと科学データは同一の配信者となっている。データにデータ処理が結び付けられており、データとデータ処理との関係は緊密である。

一方で、データ処理の利用は月光のサービスでのみに限られる。月光のデータ処理はJavaScriptでコーディングされているが、データのインターフェースなど様々な処理と一緒にまとめられた月光ライブラリとして配信されている。各データ処理を個別に取り出し、反映させることは難しい。月光以外のデータに月光ライブラリのデータ処理を反映させることはできない。

また、「月光」はスペクトルの可視化・閲覧を念頭に作られており、本論文執筆時点では、図1の「吸収帯検出」、その特徴に応じた「分類」、分類結果の「可視化」といったデータ処理を行うことができず、図1⑤で示した可視化画像を生成することができない。図1③のデータはダウンロード可能であるため、このデータを用いてその後の処理は別途行う必要がある。

	データとプログラムの 配信者	データに対する妥当な データ処理の適用	データ処理の組み換え	他のアプリケーション でのプログラム利用	他のプログラムの適用
月光	同じ	◎	×	×	×
QGIS	違う	△	×	×	○ プラグインとして
新方式	サービスとして 分けるべき	◎	◎	◎	◎

表 1 データ処理プログラムの比較表

3.2 データ提供者以外からデータ処理プログラムを受け取る事例：QGIS のプラグイン

QGIS¹⁴⁾ はフリーで利用できるデスクトップ型 GIS アプリケーションである。QGIS には標準の機能以外にも、GIS で一般的に用いられる処理について、多くのプラグインが有志により開発されている。それらのプラグインは QGIS リポジトリからダウンロードすることによって利用できる。

QGIS のプラグインは、図 4(b) に示したとおり、データの配信者とデータ処理プログラムの配信者は異なる。このためデータとデータ処理との関係は希薄である。ユーザがあるデータ処理を本来適用すべきでないデータに反映させてしまう可能性が残る。また、提供されているプラグインは配信されるデータの処理レベルが画像データのような処理レベルが高次である場合に限定される。さらに、プラグインは QGIS でのみの利用に限られ、他のアプリケーションでの再利用はできない。

3.3 既存の利用可能なデータ処理プログラムの課題

上記のとおりユーザが利用可能なデータ処理プログラムの実例はあるが、それぞれのサービスには課題がある。表 1 にまとめを示す。

まず第 1 の課題は、データとデータ処理との関係が、ある意味、ユーザの曖昧な信頼で成り立っているという点である。データに詳しい者がデータとデータ処理の関係を保証することが必要である。データとデータ処理の配信元が同一であれば、組み合わせはまず間違いないといえる。あるいは、データ処理プログラムの配信者が、特定のデータに対して対応していると明示していれば、あやまったデータ処理の適用が防げる。

第 2 の課題は、データ処理が配信されたとしても、「月光」のように特定のアプリケーションでのみの利用に限定されると、ユーザ自ら用意したプログラムの適用ができない、プログラムの再利用ができないといった問題が生じる点である。データ処理は特定のサービスやアプリケーションから独立であるべきであると考えられる。

4. 提案するプロトコル

これまでの検討を軸にして、既存の方式の課題を解決するための基本概念を整理し、再考すると以下にまとめられる。

1. データとデータ処理は関連付けられるべきである。
2. データ処理はモジュールとして組み合わせ可能な粒度が望ましい。
3. データ処理プログラムは特定のアプリケーションから自由であるべきである。
4. データ処理の組み合わせは、パッケージとしてまとめられるべきである。
5. 配信側の一連のデータ処理はユーザ自らの操作を可能とするためパッケージとして明示するべきである。
6. パッケージはデータ処理のガイドラインという位置づけであり、ユーザによるパッケージ内容の変更を可能にすべきである。

事項以降では、これらの基本概念について検討をおこなう。

4.1 基本概念 1：データとデータ処理プログラムの関係

前章で指摘したとおり、既存のデータ処理プログラムの配信では、データとデータ処理プログラムの関係は明示されているわけではなく、ユーザによる曖昧な信頼に限られていた。しかしながら、信頼に頼ると、プログラムの配信者が想定していなかったプログラムが誤って利用される可能性がある。これは、可視化された画像やグラフについて、それぞれの図がどのように作られているかプログラムレベルで明示されていないために生じる。

図 5 に示したように、データとデータ処理の関係性を明示的に示したメタデータを配信すれば、他者の検証が可能になり、より適切なデータへのデータ処理の適用が可能になるといえる。パッケージが対象とするデータとデータ処理によって得られるアウトプットを明示することが重要となる。

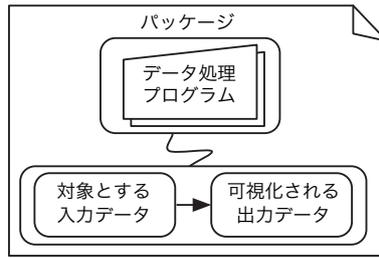


図 5 データとデータ処理の関係を示したメタデータの概念図

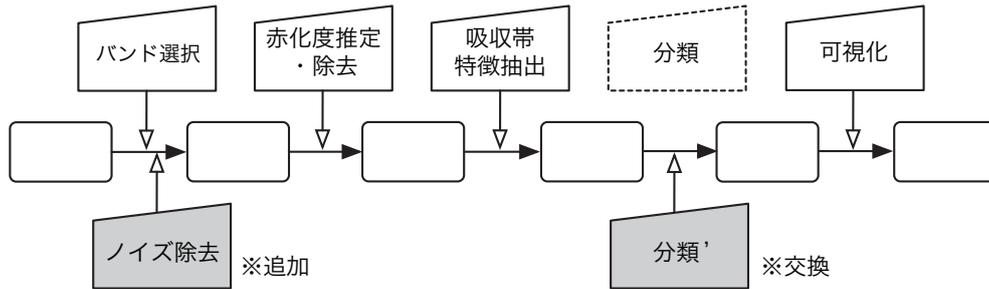


図 6 ユーザによるデータ処理への関与

4.2 基本概念2：望ましいデータ処理プログラムの粒度

データ処理プログラムは単に配信されれば良いというわけではなく、配信処理の構成自体に工夫が必要である。

図1で示したデータ処理について、例えばただ単にひとつかたまりのプログラムとして配信したならば、ユーザのデータ処理への関与は、単にそのひとつかたまりのデータ処理を実行するかしないかの選択肢に限られてしまう。一方、データ処理の粒度でプログラムを配信すれば、ユーザはデータ処理の粒度でデータ処理に関与できる可能性が生まれる。

前者も後者も得られる結果は同じだが、後者ではユーザがデータ処理に関与することによって得られる結果に変化を加えることができる点異なる。

図6の灰色で塗りつぶしたデータ処理は図1からの変更であり、新たに「ノイズ除去」を追加したり、「分類」を「分類'」に交換している。適切な粒度でデータ処理のプログラムを配信すれば、様々なデータ処理を自由に組み立てられる。ユーザの関与の仕方得られる可視化データは多様に変化する。このデータ処理のプログラムは実際には数値計算の関数によって構成される。

4.3 基本概念3：オープンなデータ処理プログラム

データ処理プログラムは、他のアプリケーションからも実行可能にするためオープンにするべきである。

特定のアプリケーションに組み込んであるデータ処理は、そのアプリケーションでのみの利用に限られ、オープンでない。対話型プログラムやGUIアプリケーションが該当する。一方で、OSのターミナル上で動作するコマンドラインや実行可能なライブラリは、ユーザが直接実行することができることはもちろん、他のアプリケーションからも実行可能であり、オープンである。

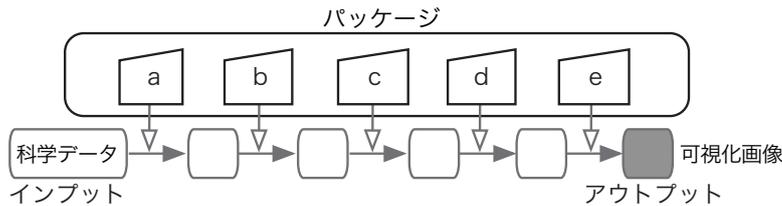
コマンドラインのデータ処理は、ユーザにとっては馴染みがないかもしれない。そこで、オープンなデータ処理プログラムに対して、GUIでラッピングする利用が考えられる。この方法であれば、データ処理のユーザビリティを保ちつつ、オープンなデータ処理を担保できる。参考になるのはGISアプリケーションのGRASS GIS¹⁵⁾である。GRASS GISはコマンドラインでの操作が基本だが、クロスプラットフォームなGUIツールキットのwxPythonを用いたGUI環境を公式に提供している。また、GUIベースのGISアプリケーションであるQGISから、GRASS GISの機能呼び出すこともできる。

4.4 基本概念4：一連のデータ処理プログラムのパッケージ

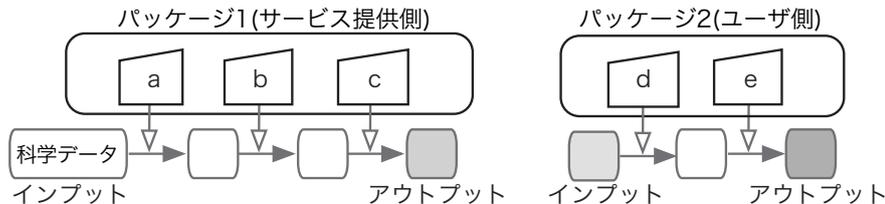
4.2のような粒度で構成されたデータ処理プログラムがあるとして、このプログラムだけでは実用的ではない。図1に示したとおりプログラムを連続的に操作することによって科学データから可視化された情報を得ることができる。このことからデータ処理プロセスを個々に配布することはもちろん、図7(a)に示したようにデータ処理のフローをパッケージとして配信することも必要である。

その際に、パッケージとしてまとめられたデータ処理のフローは図7(b)のとおりサービス提供側とユーザ側のそれぞれで利用が可能である。サービス提供側は、いかにデータ処理がなされたかを明示する目的でパッケージを利用できる。詳細は

(a) 科学データから可視化までの一連のプログラムを含んだパッケージ



(b) サービス提供側とユーザ側とに分割されたプログラムのパッケージ



(c) プログラム群のパッケージの利用例

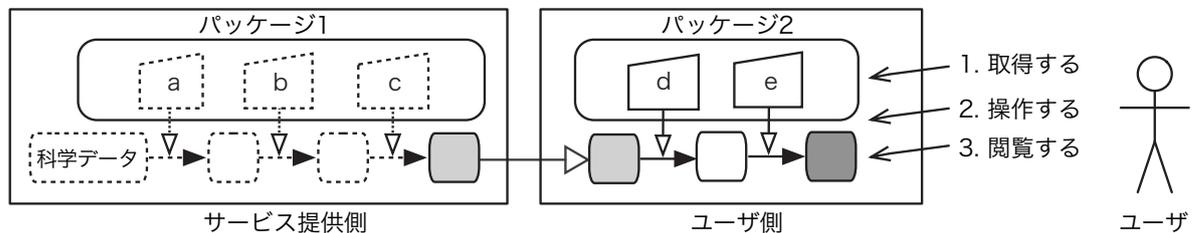


図 7 パッケージ化されたデータ処理とその適用

4.5 で述べる．ユーザは、受信したデータに対してデータ処理を適用する目的でパッケージを利用できる．詳細は 4.6 で述べる．この場合、図 7(c) に示したように、配信側とユーザ側の両者にわたって、データ処理を明示的に示すことが可能となる．

4.5 基本概念5：配信されるデータのリバース・エンジニアリング

配信されるデータがどのように作成されたかを示すことは、配信データの検証およびデータリテラシーにおいて重要である．どのようなユーザであっても初めて遭遇するデータに対しては未知の部分が多い．配布されるデータが高次であったとしても、データのリバース・エンジニアリングが可能であれば、そこをきっかけにデータに対するオペレーションに詳しくなることが可能である．

ユーザは、どのようにデータ処理がなされているかを確認するところからデータ処理の関与を始めることができる．次に、上記に述べたデータ処理の組み換えができるようになるだろう．さらに、上級者になれば特定のデータ処理に対してプログラミングを行い、そのプログラムを適用することが可能である．

4.6 基本概念6：ユーザによるデータ処理の組み換え

パッケージはそのままデータに対して適用してもよいが、ガイドライン的な位置づけとして利用することもできる．図 6 に示したように、ユーザはパッケージ中に示されている内容を参考にしつつ、データ処理をひとつずつ追いかけることができる．また、ユーザは必要に応じてデータ処理プログラムの追加や交換などの操作を自由に行うことができる．ユーザにとってデータ処理の敷居がさがることにつながる．

4.7 新方式の可能性

上記の基本概念を踏まえてシステムの実装を行うならば次に示す可能性が見えてくる．

第一に、データに対してどのような操作をするべきかわからないユーザであったとしても、可視化済みのデータを閲覧するのと変わらない体験を得ることができる．

第二に、ユーザの能力次第で、データのもつポテンシャルを引き出すことが可能となる．配信されているデータ処理に加えて、ユーザ自らプログラミングしたデータ処理プログラムも組み合わせた処理フローが可能である．データ処理に長けたユーザは自らが必要と考えるデータ処理のみの開発に集中することができる．

第三に、データ処理をオープンソース化し共有化することによって、ユーザはデータ処理をソースコードのレベルで理解することが可能となる．また、様々な外的な評価を受けることによって、データ処理の信頼性を高めることができる．

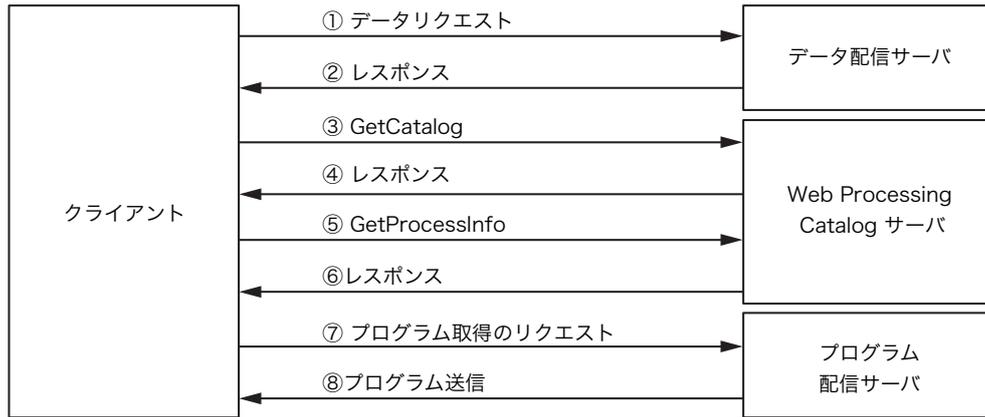


図 8 仮実装したサービスのシーケンス

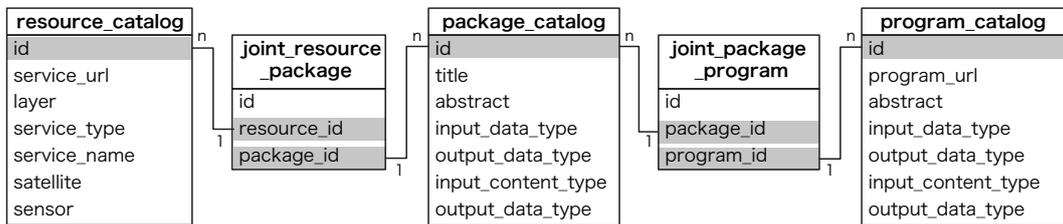


図 9 WPC メタデータを格納するデータベースの構造

本論の事例では、ひとつのデータリソースに対するデータ処理について検討をおこなってきた。実際の一連のデータ処理では複数のデータの組み合わせによって最終的に閲覧されるデータの生成が多く行われる。その場合は無限のデータ処理の組み合わせが考えられ、プロセス共有の重要性も高まるといえる。

5. 試験的な実装

本研究では前章までの議論を踏まえ、試験的にデータとデータ処理を結びつけるシステムの試験的な構築を行った。システムの実装にあたり、既存の利用できる情報システムは手を入れずそのまま利用することを心がけた。実用化に至った場合、既存の情報システムをそのまま活かすことによって、本研究に基づいたシステムの導入が可能となる。

5.1 システムの概要

本論では、データとデータ処理の関係性を示すメタデータの配信サービスを担うプロトコルを、仮に Web Processing Catalog(WPC)と呼ぶこととする。また、WPCを用いて配信するメタデータはWPCメタデータ、WPCメタデータを配信するサーバはWPCサーバと呼ぶこととする。

仮実装したシステムにおけるクライアントとサーバのサービスフローを図8に示す。仮構築を行ったシステムは、データ配信サーバ、WPCサーバ、プログラム配信サーバ3つのサーバから構成されており、これら3つのサーバが提供するサービスはそれぞれ独立している。データ配信サーバはWMSやWFSの通信が可能な既存のMapServerを利用した。また、プログラム配信サーバはHTTPの通信が可能な既存のApacheを利用した。WPCサーバについては、既存のものがないため、今回新たに実装を行った。WPCサーバとクライアントとのデータ通信は、WMSやWPS同様、地理情報の配信で多く用いられるRESTを採用した。

WPCメタデータは、WPCサーバ上のリレーショナル・データベースSQLiteで管理しており、そのデータ構造を図9に示した。図9は、図8のサービスフローで必要となるデータの順序にならい、左から右へテーブルを配置した。テーブル「resource.catalog」には、配信されているデータの情報源に関する情報を格納している。テーブル「package.catalog」には、プログラム群で構成されるパッケージに関する情報を格納している。パッケージのプログラムを動作させるにあたり必要なデータのインプットとアウトプットの情報を含んでいる。テーブル「program.catalog」には、プログラムのメタデータを格納している。プログラムの参照情報のみで、実際のプログラムコードは含んでいない。それぞれのテーブルはN:Nで結合している。

また、図10で示したとおり、WPCメタデータはXMLで記述した。これは機械可読を狙ったものであり、例えばクライアント側におけるWPCとのやりとりの自動化や、データ処理プログラムの自動取得、データ処理プログラムの自動実行など

(a) GetCatalog 図8のシーケンス③④

http://192.168.100.3/cgi-bin/wpc.cgi?REQUEST=GetCatalog&
Server=http://test_server/cgi-bin/mapserv?/moon/kaguya_sp_wfs.map&Layer=kaguya_sp

-<Web_Processing_Catalog version="0.0.0">

<pre> -<Service> <Request> GetCatalog </Request> <Provider> CAIST, The University of Aizu </Provider> </Service> </pre>	WPC情報	
<pre> -<DataResource ResoucelID="1"> -<Server> http://test_server/cgi-bin/mapserv?/moon/kaguya_sp_wfs.map </Server> <Layer> kaguya_sp_wfs </Layer> <ServiceType> WFS </ServiceType> </DataResource> </pre>	<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; padding: 2px;">データ配信 サーバURL</div> <div style="border: 1px solid black; padding: 2px;">レイヤー情報</div> </div>	指定した情報 源に関する 情報
<pre> -<PackageList> -<Package PackageID="1"> <Title> pickup_recommended_data </Title> <abstract> Pickup recommended data. </abstract> </Package> -<Package PackageID="2"> </pre>	パッケージ 概要	パッケージ群 情報

(b) GetProcessInfo 図8のシーケンス⑤⑥

http://192.168.100.3/cgi-bin/wpc.cgi?REQUEST=GetPackageInfo&PackageID=7

-<Web_Processing_Catalog version="0.0.0">

<pre> -<Service> <Request> GetPackageInfo </Request> <Provider> CAIST, The University of Aizu </Provider> </Service> </pre>	WPC情報	
<pre> -<Package PackageID="7"> <Title> get_absorption_band </Title> -<Abstract> Detect the absorption bands and describe the parameters characterizing the band shapes. </Abstract> <Input_data_type> array[float] </Input_data_type> <Output_data_type> float </Output_data_type> <Input_content_type> wavelengths </Input_content_type> <Output_content_type> wavelength </Output_content_type> <Input_Resource_id> 1 </Input_Resource_id> </Package> </pre>	<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; padding: 2px;">パッケージ 概要</div> <div style="border: 1px solid black; padding: 2px;">入出力に 関する情報</div> </div>	パッケージに 関する情報
<pre> -<ProgramList> -<Program ProgramID="1"> -<Program_URL> http://test_server/resource/pickupRecommendedData.py </Program_URL> </Program> -<Program ProgramID="3"> -<Program_URL> http://test_server/resource/continuum.py </Program_URL> </Program> -<Program ProgramID="7"> </pre>	プログラム 情報	プログラム群 情報

図 10 Web Processing Catalog のメタデータとその構造

が可能になる。どのアプリケーションからでも利用可能とするためには、XML のデータ構造を明確にする必要があり、XML Schema などのスキーマ言語による定義が必要である。今回の実装は仮実装であるため、スキーマ言語は用いなかった。

図 2⑥「可視化画像」のスペクトル特徴量マップは、実装した WPC サーバから取得した WPC メタデータ (図 10) , および WPC メタデータを頼りに取得したデータ処理プログラムを用いて解析し作図した。

5.2 システムのシーケンス

ユーザはまずデータ配信サーバにデータのリクエストを出し ①, データを受け取る ②. これは既存の仕組みの部分であり、データファイルの受け取りに該当する。

そして、データ処理プログラムの取得に移る。まず取得したデータに対してどのようなデータ処理のプログラムが存在するか WPC サーバへ問い合わせる ③. 配信データの特徴は、図 10(a) で示した URL を用いて HTTP リクエストで行う。ここでは「REQUEST」「Server」「Layer」の項目を用いている。「REQUEST」では「GetCatalog」を指定しているが、これはデータ処理の存在を WPC サーバへ問い合わせる命令である。「Server」はデータを配信しているサービスの URL 「http://test_server/cgi-bin/mapserv?/moon/kaguya_sp-wfs.map」を、「Layer」はデータ配信サーバのレイヤ「kaguya_sp」を指定している。

もしデータに対応したデータ処理のプログラムがあれば、ユーザは図 10(a) のような XML で記述されたリストを受け取ることができる ④. 図 10 に示した XML は、Firefox により整形された XML になる。

「server」で指定した値はデータベースの「resource_catalog」テーブルの「service_url」, 「layer」で指定した値はデータベースの「resource_catalog」テーブルの「layer」と照合し、適合するデータレコードがあれば、「package_catalog」のレコードのデータを送り出す。

ユーザは図 10(a) のリストの中から希望するデータ処理群のパッケージを探し出し、WPC サーバへ問い合わせる ⑤. 問い合わせには、XML 中の「PackageID」を用い、図 10(b) で示した URL を用いて HTTP リクエストで行う。ここでは「REQUEST」「PackageID」の項目を用いている。「REQUEST」では「GetPackageInfo」を指定しているが、これはパッケージおよびパッケージに含まれるデータ処理プログラムの情報を得るため WPC サーバへ問い合わせる命令である。「PackageID」は、「7」を指定している。

すると、ユーザは図 10(b) のような XML で記述された情報を受け取ることができる。この XML には、パッケージの概要や、パッケージを適用する際に必要な入出力情報、パッケージのプログラムの配信先の情報が含まれている。

そこでユーザは XML からプログラムを取得できる URL を用いて、プログラム配信サーバへ問い合わせ ⑦, プログラムを受け取る ⑧ことができる。

その後、「GetPackageInfo」の情報に基づき、プログラムの実行を行うことになる。「GetPackageInfo」は XML で記述されており、ユーザ側のアプリケーションで XML をパースした後、アプリケーションプログラムの自動実行も可能である。

6. 課題

本研究ではデータ処理の配信の仕組みについて記述したが、実際のサービス提供に向けて解決しなければならない課題がある。

まず、データ処理のプログラム群の実装だが、例示したとおり試験的に SP のデータ処理について Python で実装したのみである。この仕組みが実用的になるためには、惑星探査で用いられる一般的なデータ処理プログラムの実装が必要である。仮実装では Python を用いたが、どのようなプログラミング言語を用いるべきか検討が必要である。プログラムの利用に関して、データの入出力のインターフェースも同時に検討が必要である。

また、データとデータ処理を関連付ける WPC メタデータについて、実際の惑星科学の状況に照らし合わせて、今回、仮実装したデータ構造でよいのか、あるいは項目は過不足ないかの検討が必要だろう。

さらに、ユーザはデータ処理の配信を受けることができユーザ自らデータ処理ができるとしたが、具体的な方法は明示していない。基本的にはデータ処理のフローに従ってデータ処理を適用すれば良いが、利便性向上のためデータ処理を担う GUI の実装が必要だろう。

謝辞

本研究は JSPS 科研費 26400458 および 25287114 の助成を受けたものである。

参考文献

- 1) 宇宙航空研究開発機構. かぐや (SELENE) データアーカイブ. <https://12db.selene.darts.isas.jaxa.jp/> (2016 年 1 月 21 日アクセス) .

- 2) School of Earth & Space Exploration, Arizona State University. Lunar Reconnaissance Orbiter Camera; WAC Global Morphologic Map. http://wms.lroc.asu.edu/lroc/view_rdr/WAC_GLOBAL (2015年10月1日アクセス) .
- 3) 林洋平, 小川佳子, 平田成, 寺園淳也, 出村裕英, 松永恒雄, 山本聡, 横田康弘, 大竹真紀子, 大嶽久志. かぐや搭載スペクトルプロファイラデータ分析のためのウェブ地理情報システム月光. 宇宙科学情報解析論文誌, Vol. 4, pp. 91-103, 2015.
- 4) 会津大学・先端情報科学研究センター. 月光. http://fructus.u-aizu.ac.jp/gekko_info/ (2015年10月1日アクセス) .
- 5) 宇宙航空研究開発機構. 月周回衛星「かぐや (SELENE)」 - 観測ミッション - TC,MI,SP. http://www.kaguya.jaxa.jp/ja/equipment/tc_j.htm (2014年5月1日アクセス) .
- 6) Matsunaga T., M. Ohtake, J. Haruyama, Y. Ogawa, R. Nakamura, Y. Yokota, T. Morota, C. Honda, M. Torii, M. Abe, T. Nimura, T. Hiroi, T. Arai, K. Saiki, H. Takeda, N. Hirata, S. Kodama, T. Sugihara, H. Demura, N. Asada, J. Terazono, and H. Otake. Discoveries on the lithology of lunar crater central peaks by selene spectral profiler. *Geophysical Research Letters*, Vol. 35, No. 23, 2008. DOI:10.1029/2008GL035868.
- 7) 杉本紘平, 林洋平, 小川佳子, 平田成, 寺園淳也, 出村裕英, 松永恒雄, 山本聡, 横田康弘, 大竹真紀子, 大嶽久志. かぐやスペクトルプロファイラデータを動的に解析するためのウェブアプリケーション開発. 2014年日本地球惑星科学連合大会, 2014年5月.
- 8) Open Geospatial Consortium. Web Map Service — OGC. <http://www.opengeospatial.org/standards/wms> (2016年1月24日アクセス) .
- 9) United States Geological Survey. Webservices - Astrodocs. <http://astrodocs.wr.usgs.gov/index.php/Webservices> (2016年1月21日アクセス) .
- 10) California Institute of Technology Jet Propulsion Laboratory. NewMoon, WMS Server for Moon data. <http://onmoon.jpl.nasa.gov/> (2016年1月21日アクセス) .
- 11) Open Geospatial Consortium. Opengis web map service (wms) implementation specification (version: 1.3.0), 2006. Reference number 06-042. http://portal.opengeospatial.org/files/?artifact_id=14416.
- 12) Open Geospatial Consortium. Web Coverage Service — OGC. <http://www.opengeospatial.org/standards/wcs> (2016年1月24日アクセス) .
- 13) Open Geospatial Consortium. Web Processing Service — OGC. <http://www.opengeospatial.org/standards/wps> (2016年1月24日アクセス) .
- 14) QGIS Development Team. QGIS; A Free Open Source Geographic Information System. <http://qgis.org/> (2016年1月28日アクセス) .
- 15) GRASS Development Team. GRASS GIS; Bringing advanced geospatial technologies to the world. <https://grass.osgeo.org/> (2015年10月1日アクセス) .

教育用計算機の利用を想定した
3次元データの簡便な可視化手法の提案—gnuplotの利用

野口克行*¹

A method of visualization of three-dimensional data by “gnuplot”

by

Katsuyuki Noguchi*¹

Abstract

We have proposed a simple and easy method plotting three dimensional data which are frequently appeared in space and Earth science fields by using “gnuplot”, which is a free and one of the popular software for plotting data and is usually installed on most of the educational computer systems in universities and institutes. As the data should be reformed into appropriate formats before the input for gnuplot, we describe how to reform the data in this paper. We show a few examples of the applications of the method proposed.

Keywords: three-dimensional data, visualization, plot, gnuplot

概要

宇宙・地球科学において、3次元データを可視化する機会が頻繁にある。このような場合、第1変数と第2変数で2次元平面を取って第3変数の値を等高線（コンター）や色で表すことが多い。3次元データを可視化可能なソフトウェアは数多く存在すると思われるが、無償かつ初心者でも容易に操作できるものとなると多くはない。本論文では、ほぼ全ての教育用計算機にインストールされている（または容易にインストールが可能）と思われる gnuplot を利用し、できるだけ簡便に3次元データを可視化できるような手法を提案する。具体的には、gnuplot の入力に適うようなデータフォーマットの変換方法を説明する。応用例としては地球および惑星大気科学分野のデータを挙げるが、本論文で示すプロット手法の原理自体は自然科学の幅広い分野のデータに応用できるものである。

キーワード: 3次元データ 可視化 プロット gnuplot

1 はじめに

近年、ハードウェア・ソフトウェア両面から、宇宙・地球観測分野のデータの質と量が劇的に向上しつつあり、膨大なデータの蓄積が進んでいる。一方で、そのようなデータを詳細に解析する際には、人間が実際にデータをプロットして可視化することが必要である場合が多い。そのため、大量のデータに対応するには、データの可視化の負担をできるだけ小さくし、効率よく、安易かつ低コストでプロット

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

¹ 奈良女子大学大学院自然科学系 (Faculty of Science, Nara Women's University)

できる環境の構築が望ましい。

大学等の教育現場においては、UNIX (Linux) をベースとした計算機環境が整備されていることが多く、そのような計算機環境においてデフォルトで動作する可視化システムがあれば、高価なワークステーションやライセンス料の高いソフトウェアを用いること無しに学生が気軽にそのようなデータに初等学年から親しむことが可能になる。

gnuplot は、そのような条件に合致するデータプロット用ソフトウェアと言える。ユーザとのインタラクティブなコマンド入力システムを提供する一方で、スクリプト化にも対応しており、初等利用者・高度利用者共に満足し得るソフトウェアの一つである。インストールに関しても、多くの Linux ディストリビューションにおいてパッケージ化されているため簡便であり、また Windows や Mac OS でも利用可能である。マニュアルは、全て gnuplot の Web ページ (<http://www.gnuplot.info/>) から利用可能である¹⁾。

一般的に、3次元データのプロットは2次元データとは異なり工夫が必要である。例えば、ある高度面における気温の経度・緯度分布をプロットする場合は、第1変数(x)を経度、第2変数(y)を緯度、第3変数(z)を気温とし、x-y平面上にzの値を色や等高線などを用いて表現することが多い。本論文では、「色(あるいは白黒の濃淡等)による塗りつぶしを用いて、3次元データを2次元面上に表現すること」を(3次元データの)可視化と呼ぶことにする。このような図のプロットは、データの格子化や欠損値の対応等が必要となり、初学者には困難なことが多い。本論文では、gnuplot マニュアル^[1]を参考とし、3次元データの可視化をできるだけ簡易かつ元のデータに忠実になるようにgnuplotで行なう手法を示す。なお、ここで示す手法は、著者の所属する奈良女子大学において学生の計算機演習などで実際に教えているものである。

2 gnuplot による簡易な3次元データ可視化の手法

gnuplot では、pm3d という機能を用いて、以下のようなコマンド列を打つことで3次元データファイル(例としてファイル名を data.dat とする)をプロットすることが可能である。

```
gnuplot> set pm3d
gnuplot> set view map
gnuplot> splot "data.dat"
```

gnuplot による3次元データの描画原理を図1に示す。gnuplot は、与えられたデータセットにおいて隣接する4点を探し出し、4点の平均値等(後述)に相当する色を用いて、4点を結ぶ矩形に塗りつぶすように描画を実行する。3点では描かれない。この仕組みを利用して、扱いやすさと正確さのバランスの取れた描画をユーザの望む形で実現することを考える。

まず、単純にこの gnuplot の描画機能を利用することを考える。最低で4点揃っていれば良い訳であるから、例えば、座標が等間隔で並ぶ格子状のデータを入力すれば、gnuplot は各格子点のデータを4点ずつ用いながら平均し、その平均値に対応する色で各点を結ぶ正方形・長方形に塗りつぶす。これは、gnuplot に自動的に各点を補間させているイメージである。そこで、この方法を「gnuplot の自動補間機能を用いる方法」と呼ぶことにする。

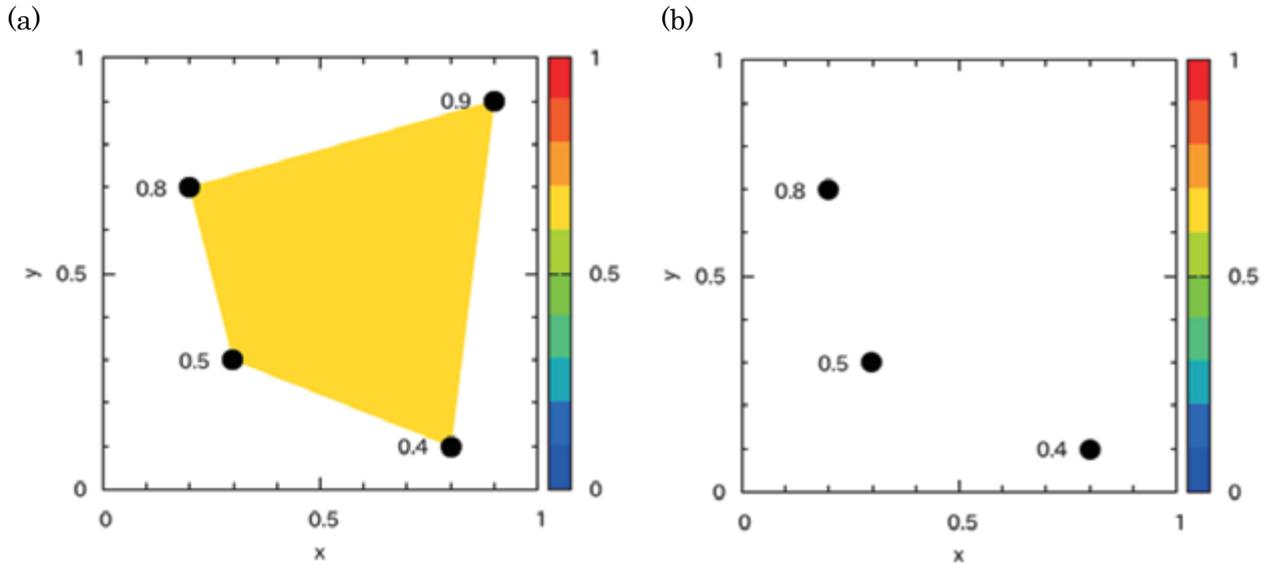


図1 gnuplotによる3次元データ描画の原理。x-y平面上に各点をプロットし、zの値を色で表現する。図中の数字は、各点のzの値である。(a)4点の場合。z値の平均は0.65 $(=(0.4+0.5+0.8+0.9)/4)$ となるが、これは黄色に相当する。この色で4点を結ぶ矩形上に塗りつぶされる。(b)3点の場合(このとき、色の塗りつぶしは実行されない)。

一方で、gnuplotに勝手に補間させるのではなく、あるデータ値を持つ領域に対して、その位置においてそのデータの値を厳密に反映した色で描画したい場合も存在するであろう。例えば、データ点が1点しかない場合、上述のgnuplotの補間機能を用いる方法ではプロットがなされない。このような場合は、例えばその点の周囲を囲むように4点取り、その4点を結んでできる矩形領域をzの値に対応する色で塗りつぶせば良いことになる。これは、データ点を一点ずつ有限の大きさにピクセル化することに相当する。そこで、この方法を「データ点をピクセル化する方法」と呼ぶことにする。

2.1 gnuplotの自動補間機能を用いる場合

gnuplotの自動補間機能を用いて3次元データを可視化する場合には、事前にデータを整形することが必要となる。具体的には、第1変数(x)についてまずソートし、次に第2変数(y)についてソートをする。その際、xが次の値に変わる前に空行を一行入れる。例として、付録1に示すようなデータセットをプロットすることを考える。このデータを整形して図2に示すようにソート・空行挿入を施してからgnuplotに読み込ませると、gnuplotは図3に示すように隣接する4点のz値を単純平均(デフォルトの場合、corners2colorオプションで変更可能)してから描画する。このようなデータ整形は、Linuxの各種コマンド(sortやawk等)を用いればそれほど手間を掛けずに実現が可能であるので、格子状データを簡易的にプロットするには実用的な方法である。ただし、次に述べるような注意が必要である。

まず、データ点数が少ない場合やデータ値の変動が激しい場合は、gnuplotによって補間された際に実際のデータの様子とは大きく異なったイメージの図が描画される可能性がある。図3の例においても、実際のデータにはカラーバーの最小値(0:濃紺)と最大値(1:赤)に相当する値が含まれているが、それらに対応する色はプロットには現れていない。データ値の変動が激しい場合には、値が平滑化され、

(a)	(b)
x y z	x y z
0.5 0.5 0.05	0.5 0.5 0.05
0.5 1.5 0.25	0.5 1.5 0.25
0.5 2.5 0.45	0.5 2.5 0.45
0.5 3.5 0.65	0.5 3.5 0.65
1.5 0.5 0.15	
1.5 1.5 0.35	1.5 0.5 0.15
1.5 2.5 0.55	1.5 1.5 0.35
1.5 3.5 0.75	1.5 2.5 0.55
2.5 0.5 0.25	1.5 3.5 0.75
...	
...	2.5 0.5 0.25
...	...

図 2 格子データの整形の例。(a) 元のデータ、(b) 整形後のデータ。第 1 変数 (x)、第 2 変数 (y) の順にソートし、x の値が変わる際には空行を 1 行入れる。

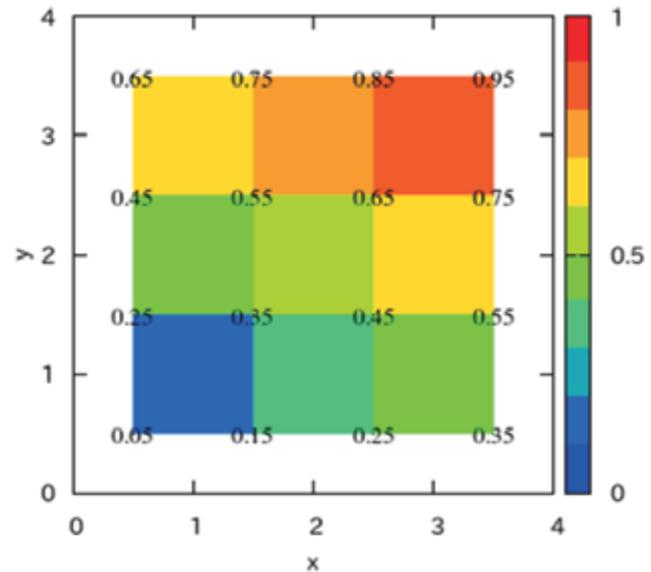


図 3 gnuplot の補間機能を用いたプロットの例。図 2(b)のデータを用いている。図中の数字は、各点の第 3 変数 (z) の値である。

元データの変動が十分に表現されない可能性がある。なお、先述の `corners2color` オプションを用いると 4 点の最大値・最小値・どれか 1 点の値・メジアン値などを表示することが可能であるが、4 点全ての情報を適切に表現しきれないという意味では同じである。

また、データに欠損がある際にうまく対応できない場合があるという問題が挙げられる。データの欠損には二通りあり、ある値で欠損値が定義されている場合 (gnuplot は、任意の文字列を欠損値として定義し、この値をプロットの対象から除外することが可能である) と、座標ごと欠落している場合とが考えられる。例として、付録 1 に示す欠損のあるデータをプロットした結果を図 4 に示す。図 4(a)のように欠損値が定義されている場合は、その点の周辺では矩形を作らずに色で塗りつぶされない。図 4(b)のように座標ごとデータから欠落している場合は、gnuplot はその周囲にある点を変則的に 4 点ずつ結ぶことで色を塗りつぶそうとする。これが自動的に補間された結果となるが、このようなプロットが必ずしもユーザの望むことかどうかは難しいところであろう。欠損値が定義されている場合でも、たった 1 点の欠損値に対して 4 つのピクセル分のデータがプロットされないことになるため、欠損が多い場合には多くのデータがプロットされないことにもなり得る。

したがって、gnuplot の自動補間機能を用いる際には、データが格子状に揃っており、データ点数がある程度多くて、値の変動が激しくなく、欠損値も無いかごく少ないことが望ましいといえる。特に、座標ごと欠落している場合は自動的に不規則に塗りつぶされる可能性があるため、注意が必要である。

2.2 データ点をピクセル化する方法

前節の gnuplot の自動補間機能を用いる方法では、必ずしも元データを忠実に描画することができない場合があり、またデータ欠損があるとユーザが想定しないような図が得られる恐れがあった。そこで、

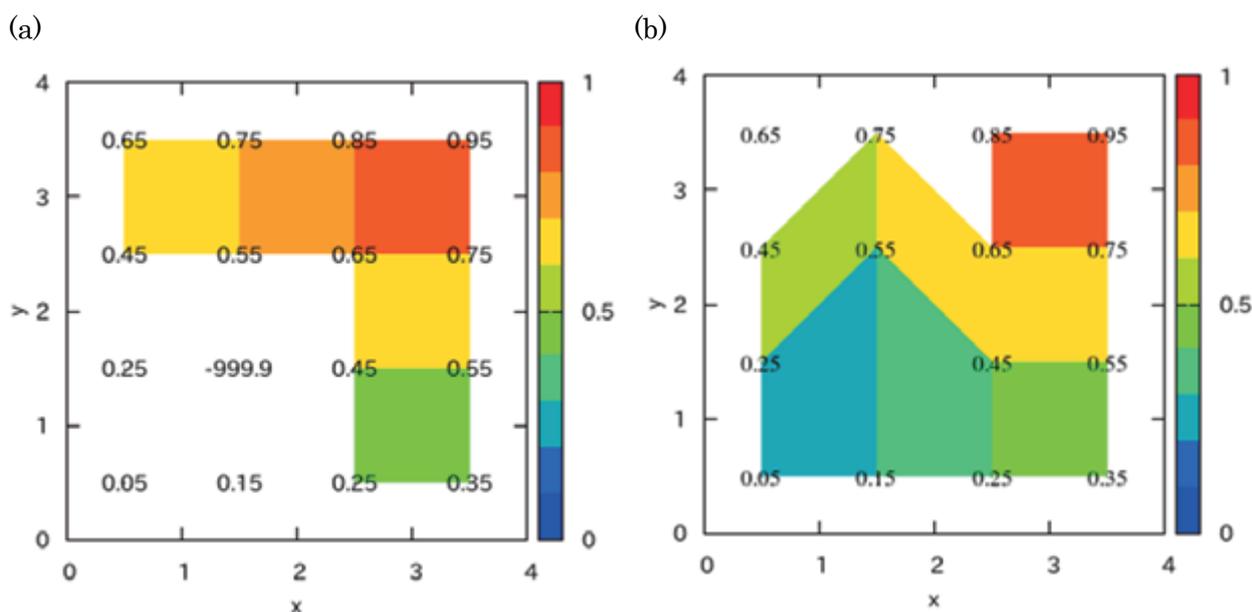


図 4 gnuplot の補間機能を用いた場合の、欠損があるデータのプロットの例。(a) 欠損値が定義されている場合 (例では-999.9)。(b)座標自体がデータに存在しない場合。図中の数字は、各点の第 3 変数 (z) の値である。(a)と(b)のいずれにおいても、座標(1.5,1.5)のデータが欠損になっている (付録 1 のデータを参照)。

データに忠実に描画することを考える。前述したように、ある 1 点のデータに対し、その点を囲むような形でピクセルを描けばよい。図 5 に、このようなデータ変換の方法の例を示す。この例では、ユーザが予めピクセルの二辺の長さ (L_x と L_y) を固定値として与えて、元のデータ点をその大きさのピクセルに変換するようにしている。データ 1 点当たり 4 点のデータを生成し、ピクセルの下側の 2 点を最初に記述した後に 1 行の空行を入れ、次にピクセルの上側の 2 点を記述して、2 行の空行を入れる。もし、ピクセルの辺の長さを個々の点毎に変えたければ、データ列をもう 2 列増やし、各データ点に L_x と L_y の値を与えればよい。

このようなデータ変換の後に gnuplot に入力すると、図 6 のようなプロット結果が得られる。この結果を図 3 と比較すると、ピクセル化したほうがより元のデータに忠実にプロットされていることが分かる。

ここまでは格子状に揃っているデータを例に挙げてきたが、データをピクセル化する手法を応用すれば、必ずしも格子状ではなく不均質なピクセル形状を持つデータでもプロットが可能である。4 つの点の $x \cdot y$ 座標情報とその点に対する z 値があれば、上述の方法に準じる形で 2 点ごとに空行を 1 行ないし 2 行加えることで、そのピクセルを忠実にプロットすることが可能である。図 7 に例を示す。この例では、人工衛星による大気汚染物質 (二酸化窒素) の観測結果をプロットしている。元のデータはセンサの視野の四隅に相当する 4 点の経度・緯度情報とその視野内での二酸化窒素量を含んでおり、図 5 に準じる形でデータ変換をすることでセンサ視野ごとの二酸化窒素量をプロットすることに成功している。なお、付録 2 に、地図 (海岸線) を gnuplot でプロットする手法を示した。地図をプロットするには GMT が有名であるが、海岸線のデータさえ入手できれば、gnuplot でも比較的容易に地図をプロットすることが可能である。

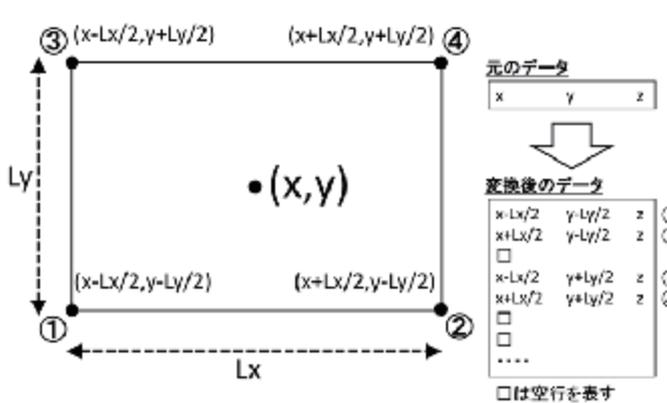


図5 gnuplot でプロットするためのデータ変換（ピクセル化）の例。元のデータ点 (x, y, z) を中心として、辺の長さが L_x と L_y である長方形を作るようにする。元のデータ 1 点当たり計 4 点を記述することになるが、最初の二つの点の組（①と②）の次には一行空行を入れ、次の二つの点の組（③と④）の後には、二行空行を入れる。

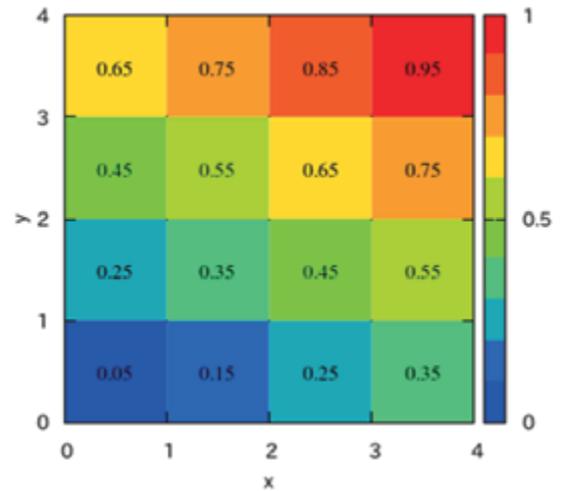


図6 データ点をピクセル化した場合のプロットの例。付録1のデータを用いている。図中の数字は、各点の第3変数 (z) の値である。

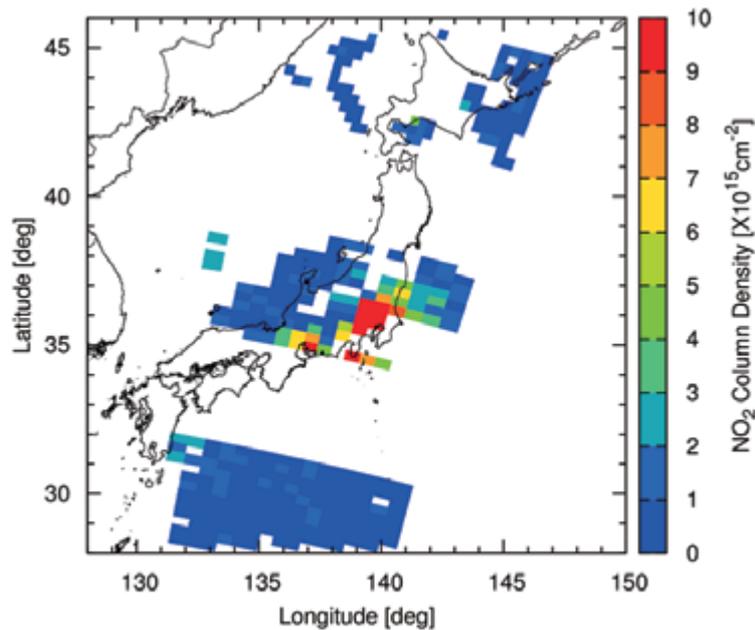


図7 格子状ではない不均質なピクセル形状をもつデータを gnuplot でプロットした例。データを取得したセンサの視野が緯度経度の格子に対して傾いており、大きさもそれぞれで異なっている。欧州宇宙機関の地球観測衛星 EINVISAT に搭載された SCIAMACHY センサによる二酸化窒素気柱量の観測結果（2009年2月7日）。地図のプロット方法は付録2を参照のこと。

2.3 画像データとしてプロットする場合との比較

これまでに述べた方法の他に、`gnuplot` では `image` オプションを用いて 3 次元データを画像データとしてそのまま値を表示する機能がある。

```
gnuplot> plot "data.dat" with image
```

この場合、データは格子状に揃っていることが必要であり、また欠損値もうまく処理することができない（つまり、付録 1 に示した 3 つのデータのうち、最初のものしかプロットできない）。また、図 7 のような不均質なピクセルサイズのデータもプロットすることはできない。そのため、数値シミュレーション結果などデータが格子状に揃っていてかつ欠損値がない場合を除けば、画像データとしてプロットするのは難しいといえる。

3 まとめ

本稿では、`gnuplot` を用いて 3 次元データを可視化する手法を提案した。簡易的に行なうには、`gnuplot` の自動補間機能を利用する。3 列の値が入ったデータファイルを用意し、第 1 変数、第 2 変数の順でソートし、第 1 変数の値が変わる毎に空行を入れる。これにより、`gnuplot` は隣接する 4 点ごとに矩形を作って第 3 変数の値を平均等した上で矩形をその値に相当する色で塗りつぶす。ただし、この方法は格子状データであること、データ数が比較的多いこと、欠損値が少ないことなどが必要である。

元のデータに対してより忠実にプロットするには、データ点をピクセル化する変換を実施してから `gnuplot` に入力する。もし、格子状のデータであれば、ピクセルとなる長方形（または正方形）の中心にデータ点があるように 4 点の座標を計算し、1 点あたり 4 行のデータとする。ただし、最初の 2 行の後に空行を 1 行入れ、最後の 2 行の後には空行を 2 行入れる。この方法を応用すれば、必ずしも格子状のデータではない不均質なピクセルサイズのデータに対しても、ユーザが 4 点の座標を与えることにより 1 ピクセルずつ忠実に描くことが可能になる。

今後ますます増大する地球・宇宙観測データに対して、容易かつ低コストで可視化を行なえる事はデータ解析を効率化する上で不可欠である。初学者にも比較的扱いやすいプロット用のソフトウェアを採用し、プロットの方法を分かりやすくかつ実践的に解説することで、データ解析の発展に寄与していきたいと考えている。

4 謝辞

図 7 の SCIAMACHY データは、ドイツ・ブレーメン大学の Andreas Richter 博士から提供頂いた。

参考文献

- 1) Thomas Williams and Colin Kelley, “gnuplot 4.4 An Interactive Plotting Program”, http://www.gnuplot.info/docs_4.4/gnuplot-ja.pdf, 2011.

付録1 本文で用いたデータ

格子状データ	欠損値定義のあるデータ	一部座標が欠落しているデータ
2.5 2.5 0.65	3.5 1.5 0.55	2.5 0.5 0.25
1.5 0.5 0.15	0.5 0.5 0.05	0.5 0.5 0.05
2.5 3.5 0.85	1.5 1.5 -999.9	3.5 2.5 0.75
3.5 1.5 0.55	0.5 3.5 0.65	0.5 2.5 0.45
0.5 0.5 0.05	3.5 2.5 0.75	1.5 0.5 0.15
1.5 1.5 0.35	0.5 1.5 0.25	3.5 3.5 0.95
0.5 1.5 0.25	1.5 3.5 0.75	1.5 3.5 0.75
3.5 3.5 0.95	2.5 0.5 0.25	2.5 1.5 0.45
0.5 3.5 0.65	2.5 2.5 0.65	1.5 2.5 0.55
1.5 2.5 0.55	0.5 2.5 0.45	0.5 1.5 0.25
0.5 2.5 0.45	1.5 2.5 0.55	2.5 3.5 0.85
1.5 3.5 0.75	2.5 3.5 0.85	3.5 0.5 0.35
2.5 1.5 0.45	1.5 0.5 0.15	2.5 2.5 0.65
3.5 0.5 0.35	3.5 0.5 0.35	0.5 3.5 0.65
3.5 2.5 0.75	2.5 1.5 0.45	3.5 1.5 0.55
2.5 0.5 0.25	3.5 3.5 0.95	

付録 2 gnuplot の 3 次元データプロット時における地図（海岸線）のプロット

gnuplot で地図（海岸線）をプロットするには、まず海岸線のデータが必要となる。例えば、米国の NOAA の Web ページ等からダウンロードが可能である。データには、緯度・経度点が含まれていればよい。実際のプロットの際には、gnuplot のコマンドプロンプトにおいて以下のようなコマンドを打つとよい。

```
gnuplot> set pm3d
gnuplot> set view map
gnuplot> splot "data.dat", "coast.dat" using 1:2:0 with line title ""
```

この例において、「data.dat」というのが 3 次元プロットを行ないたいデータファイル名、「coast.dat」が海岸線データファイル名である。海岸線データには、以下に示すように第 1 変数に経度、第 2 変数に緯度を格納し、島等の閉曲線を描くためには、閉曲線と閉曲線のデータの間に行空行を入れる。

```
114.221346    10.21988
114.221346    10.223401
. . .
114.221639    10.219587
114.221346    10.21988

114.242174    10.176757
114.242174    10.178224
. . .
114.242467    10.17617
114.242174    10.176757

114.338688    10.738237
. . .
```

「はやぶさ 2」運用補助のための可視化手法 —影の描写の検討—

三浦 昭^{*1}, 山本 幸生^{*1}, 吉川 真^{*1}

Visualization Methods to Support operations of “Hayabusa2” - A Study on rendering shadows -

Akira MIURA^{*1}, Yukio YAMAMOTO^{*1}, Makoto YOSHIKAWA^{*1}

Abstract

In this paper, we study on algorithms to render shadows precisely and quickly in scenes of the visualization software that we intend to use in operation phase of the space probe “Hayabusa2” and the asteroid 1999JU₃. By means of ray tracing algorithm, we can render shadows more precisely than those of Z-buffering. We have found that, using simplified algorithms of ray tracing and hardware accelerations of GPGPU (OpenCL), it is possible to use ray tracing to render scenes at operations where real-time responses are required. Using a hybrid method of Z-buffering accelerated by OpenGL and ray tracing accelerated by OpenCL, it is possible to render quickly enough in such situations as changing viewpoint using GUI, and so on.

Keywords: Hayabusa2, visualization, ray tracing, OpenGL, OpenCL, GPGPU, Speeding up

概要

本稿においては、「はやぶさ 2」と小惑星 1999JU₃を想定した、探査機運用者向け可視化ソフトウェアにおける、正確な影 (shadow) の描写と、レンダリング速度の向上について検討した。深度バッファを用いた手法に比べてレイトレーシングは影の描写が正確である。本稿においては、レイトレーシングのアルゴリズム簡略化と GPGPU (OpenCL) による高速化を図った結果、リアルタイム性が要求される運用においても、レイトレーシングによるレンダリングが実用的な速度で実現可能であることが示唆された。また OpenCL によるレイトレーシングと OpenGL による深度バッファを用いた計算と組み合わせることにより、リアルタイム性の高い視点変更等の操作にも耐えうるレンダリング速度が得られることが確認できた。

キーワード: はやぶさ 2, 可視化, レイトレーシング, OpenGL, OpenCL, GPGPU, 高速化

1. はじめに

本稿においては、「はやぶさ 2」と小惑星 1999JU₃を想定した、探査機運用者向け可視化手法における、高精度な影 (shadow) の描写と、レンダリング速度の向上について検討する。

筆者らはこれまで、「はやぶさ」の軌道の可視化について、その軌道の推定から各種可視化手

法まで検討して来た^{[1][2]}。リアリティを迫及した可視化手法としては、レイトレーシング⁽³⁾に基づいた CG ソフトウェアが各種提供されており、筆者らも POV-Ray^{*2}を用いた可視化を試みてきた。しかしながら POV-Ray はレンダリング速度・レンダリング機能の両面で、運用に係る要求仕様を満たすことが難しくなった。要求仕様を満たすような商用・非商用 CG ソフトウエ

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

^{*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)

^{*2} Persistence of Vision Raytracer Pty. Ltd., “POV-Ray - The Persistence of Vision Raytracer”, <http://www.povray.org/> (2015 年 11 月現在)

アについても調査を進めたものの、要求仕様に含まれると想定される Hapke モデル⁽⁴⁾等の機能を実装した製品が本稿執筆時点では見受けられない等、調査は難航している。

そこで筆者らは、要求される機能・性能を満たすような、応答性能の高い可視化ソフトウェア開発に着手した。その開発にあたって課題の一つとなったのは、影 (shadow) を高精度・高速にレンダリングすることである。探査機の運用やその模擬・訓練にあたっては、影の描写が重要となるフェーズがあり、本稿においては、要求仕様を満たすような分解能を維持しつつ、実用レベルの応答になることを一つの目安として、影に着目したレンダリング手法について検討を行う。

レンダリングの精度という観点では、画素毎に輝度計算するレイトレーシングが有利であり、今後のアプリケーションにも適用したいところであるが、速度面での優位性は低い。一方で OpenGL^{*3} と GPU の組み合わせによるプリミティブ毎の計算は、速度面では有利であるが、精度面での課題がある。

本稿においては、これらの手法を要素毎に分解し、精度の観点と速度の観点の両面から、運用に供する妥当性のある手法を検討し、評価することとした。一つには、精度の面で有利であるレイトレーシングを用いたリアルタイムレンダリングが実現可能であるかについて吟味する。その一方で、速度面でレイトレーシングを補えるよう、速度重視の手法で妥当な精度を実現できるかについても吟味する。

2. 要求仕様

現時点で探査機の運用者向け可視化ソフトウェアに対して要求されると想定される仕様を表 1 に示す。これらの仕様は「はやぶさ 2」運用関係者へのヒアリングやプレゼンテーション等に基づいたものである。可視化の用途によってはこれらの一部が緩和される場合もある。

またこれらはいずれも正式な仕様を表すものではない。表中で「(仮)」と記した箇所は、要

求仕様を待つまでの間の、仮の設定値である。

これらの内、陰 (shade)、テクスチャ、歪については将来制定される要求仕様を待つ段階である。それぞれに対応できる拡張性を意識する必要はあるが、本稿においては、形状と影 (shadow) を要求仕様に応じて描くことに注視する。

2016 年以降、模擬・訓練の時点では、模擬用の小惑星形状や影を 10cm の分解能で表示する必要がある。その際に必要とされる画像解像度は 512×512 ピクセルから 1024×1024 ピクセルである。

2018 年の運用フェーズでは、4K ディスプレイ上での GUI 操作が予定されている。表示する画像としては 1,024×1,024 ピクセルを超える可能性もあるが、本稿の時点では 1,024×1,024 ピクセルを想定して評価する。

表 1 要求仕様

主用途	模擬, 訓練	運用
開発時期	2016 年～	～ 2018 年
UI	CLI/GUI (仮)	GUI
画像解像度	512×512 1024×1024	512×512 1024×1024 その他 (仮)
分解能	10cm (精査時)	
レンダリング方法	リアルタイムレンダリング プリレンダリング	リアルタイムレンダリング
表示速度	TBD	精査: 3 fps (仮) 高速: 10fps (仮)
表示項目	小惑星 探査機及び探査機の付随物 各種補助表示 (物理量, 方向, 座標等)	
影 (shadow)	上記分解能で影を表示すること。	
陰 (shade)	Hapke モデル等, 今後制定される要求仕様に基づいて上記分解能で陰を自動生成するプログラムが可能であること。	
小惑星のテクスチャ	今後制定される要求仕様に基づいて, 上記分解能で小惑星のテクスチャを自動生成するプログラムが可能であること。	
歪み	今後制定される要求仕様に基づいて, カメラの光学系の歪み等を再現するプログラムが可能であること。	

^{*3} Khronos Group, “OpenGL - The Industry Standard for High Performance Graphics”, <https://www.khronos.org/> (2015 年 11 月現在)

本稿執筆時点の評価ソフトウェアを用いたプレゼンテーション等からの経験則により、10cmの分解能が要求されるような精査時には、毎秒3フレーム程度以上の表示能力を想定する。

また視点変更等の操作時にはマウス等による指示が迅速に反映されるよう、さらに高速の表示モードが必要と考えられる。本稿においては経験則から、高速表示モードとして10fpsを想定する。高速表示モードにおいては形状や影の精査はしないことを前提に、簡易表示を行うことも現実的な選択肢としてありうる。

「はやぶさ2」は小惑星到着時に、高度20km付近から小惑星を観測する予定となっており^[5]、その後に小惑星への降下が予定されている。そのため、可視化ソフトウェアとしては、遠方から至近距離まで破たんのないレンダリングが必要となる。また運用フェーズにおいては運用者が画像の視点・注視点等を自由に設定することが想定される。要求仕様に基づいてレンダリング手法を検討する際は、このような視点・注視点の自由度の高さも考慮する必要がある。

3. 評価に用いる環境・データ

3.1. ハードウェア・ソフトウェア環境

本稿で評価に用いるハードウェア及びソフトウェアについて、以下に示す。以後本稿においては、「Mac Pro」、「MacBook Pro」、「Mac mini」の呼称は以下の具体的な環境に対する固有名詞として使用する。

用いるフレームワークはいずれもXcode^{*4}に標準で添付されているものである。その他のAPI、ライブラリ等は自製のものを使用する。

本稿の手法を具体的に実装するプラットフォームは未定であるため、幅広いプラットフォームに適用可能なプログラム手段を採用する。検討対象としている手法自体のコーディングは、CPU側のプログラムはC++を用い、

GPUによる高速化はOpenGL^{*3}とOpenCL^{*5}を用いる。プラットフォーム依存性があるObjective C++とCocoaはベンチマーク用の初期設定を行うための利用に留める。

(a) Mac Pro 環境

筐体: Mac Pro (Late 2013)
 CPU: Xeon E5 プロセッサ 3.5GHz 6 コア
 GPU: AMD FirePro D700^{*6} 2 式
 6GB GDDR5 VRAM
 2048 stream processors
 3.5TFLOPS
 主記憶: 64GB
 OS: OS X Yosemite
 開発環境: Xcode 7.1(7b91B)
 フレームワーク
 Cocoa
 OpenGL (version 2.1)
 OpenCL (version 1.1)
 コンパイラ:
 clang (C++, Objective C++):
 LLVM version 7.0.0
 opencl (OpenCL): LLVM version 3.2

(b) MacBook Pro 環境

筐体: MacBook Pro (Retina, 15-inch, Mid 2014)
 CPU: Core i7 プロセッサ 2.8GHz 4 コア
 GPU1: Intel Iris Pro Graphics 5200^{*7} (CPU内蔵)
 1.5GB SDRAM
 40 execution units, 280 threads
 832GFLOPS @1.3GHz
 GPU2: NVIDIA, GeForce GT 750M^{*8}
 2GB VRAM
 384 unified shaders
 722.7GFLOPS
 主記憶: 16GB
 OS: OS X Yosemite
 開発環境: Xcode 6.3(6D570)
 フレームワーク
 Cocoa

^{*4} Apple Inc., "Xcode - What's New - Apple Developer", <https://developer.apple.com/xcode/index.html> (2015年11月現在)

^{*5} Khronos Group, "OpenCL - The open standard for parallel programming of heterogeneous systems", <https://www.khronos.org/opencl/> (2015年11月現在)

^{*6} Apple Inc., "Mac Pro - Technical Specifications - Apple", <http://www.apple.com/mac-pro/specs/> (2015年11月現在)

^{*7} Intel Corporation, "The Compute Architecture of Intel® Processor Graphics Gen7.5", https://software.intel.com/sites/default/files/managed/f3/13/Compute_Architecture_of_Intel_Processor_Graphics_Gen7dot5_Aug2014.pdf, pp.12-13 (2015年11月現在)

^{*8} From Wikipedia, the free encyclopedia, "GeForce 700 series", https://en.wikipedia.org/wiki/GeForce_700_series (2015年11月現在)

OpenGL (version 2.1)
 OpenCL (version 1.1)
 コンパイラ :
 clang (C++, Objective C++):
 LLVM version 6.1.0
 opencl (OpenCL): LLVM version 3.2

(c)Mac mini 環境
 筐体 : Mac mini Late 2012
 CPU: Core i7 プロセッサ 2.6GHz 4 コア
 GPU: Intel HD Graphics 4000⁹⁾ (CPU 内蔵)
 1GB SDRAM
 16 execution units, 112 threads
 256GFLOPS @1GHz
 主記憶 : 16GB
 OS: OS X Mavericks
 開発環境 : Xcode 6.2(6C131e)
 フレームワーク
 Cocoa
 OpenGL (version 2.1)
 OpenCL (version 1.1)
 コンパイラ :
 clang (C++, Objective C++):
 LLVM version 6.0
 opencl (OpenCL): LLVM version 3.2

3.2. 表示すべきオブジェクト

本稿において表示すべきオブジェクトを表 2 に示す。以後、レンダリングにあたっては、ひとまとまりの形状をオブジェクトと称し、そのオブジェクトを構成する個別の原始的な形状をプリミティブと称する。

各種物理量や補助線等は、運用者を補助するために、各々の特徴に応じたモデリングが想定される。本稿においては、その例として LRF データを扱う。LRF は探査機から幾つかの方向にビームを放って測距する仕様となっており、得られた距離とビーム方向から、小惑星表面の位置を推定することができる。本稿で用意した LRF オブジェクトは四角錐で、錐の先端が測距点を示す。

本稿においては、探査機は 1 つのオブジェク

トで表現する。小惑星のモデルは同一形状でありながら解像度の異なる 4 種類を用いるが、評価に当たっては、シーン毎に 1 つのオブジェクトを選択する。LRF データは 4 方向分を可視化するものとして、4 オブジェクト用意する。これらを合計して、1 つのシーンで同時に用いるオブジェクト数は 6 となる。

表 2 速度比較に用いるオブジェクト

表示項目	オブジェクトの概要	プリミティブ数	
		略記	総数
探査機	「はやぶさ 2」		213,405
小惑星	Gaskell イトカワ形状モデル (右記いずれかを選択)	49K	49,152
		196K	196,608
		786K	786,432
LRF	四角錐 (4 式)	3145K	3,145,728
			4×4 式

これらの形状データは、いずれも実際の「はやぶさ 2」や小惑星等を正確に再現したものではない。本稿においては、与えられた形状データに基づいて適切なレンダリングができることを確認するために、これらのデータを用いている。その範囲内においては、形状データ自体と現実の探査機等との形状の相違は本質的な問題ではない。

2016 年の模擬・訓練以降は、現実の探査機の形状を反映したレンダリングを行う必要があり、フライトモデルの形状をフィードバックすることは、今後の課題として残されている。同様に小惑星 1999JU₃ の詳細な形状モデルも得られていないため、小惑星イトカワの Gaskell 形状モデル^{*10}を借用した。2016 年以降の模擬・訓練にあたっては詳細な仮想小惑星モデルを用いることが想定される。また実運用にあたっては、随時小惑星の推定形状がアップデートされることとなる。

「はやぶさ 2」の小惑星接近時の実データはまだ取得されておらず、LRF をはじめとした、本稿で用いる位置・姿勢等の諸データは、模擬データである。

⁹⁾ David Cowperthwaite, David Hoff, “Intel® Processor Graphics: Now Serving the Cloud”, <https://software.intel.com/sites/default/files/managed/a6/e3/Now%20Serving%20the%20cloud%20Cowperthwaite%20.pdf>, p.6 (2015 年 11 月現在)

^{*10} HAYABUSA PROJECT SCIENCE DATA ARCHIVE, “Shape Model”, <https://darts.isas.jaxa.jp/planet/project/hayabusa/shape.pl> (2015 年 11 月現在)

4. レンダリング手法

要求仕様の中で、プリレンダリングについては、時間的な制限が緩やかであるため、レイトレーシングを用いるものとする。一方でリアルタイムレンダリングについては、精度と時間的な制約とのトレードオフを検討する必要がある。本章においては、OpenGL等で多用されているプリミティブ毎のレンダリング手法と、レイトレーシングに代表される画素毎のレンダリング手法とを、要素に分解して整理する。

探査機や小惑星の陰 (shade) は、本稿における評価ではランバート反射によるフラットシェーディングを用い、テクスチャについては単色とする。光学系の歪みは考慮しない。これらは、係る要求仕様が定まった時点で検討・評価するものとする。

異なる手段においても、ソースコードや座標変換等の手順を共有できるように、パイプラインの各段における、座標変換、視点変換、クリッピング等に係る行列の定義はOpenGLで用いられる行列に揃え、各種座標変換に係る配列要素の計算方法もOpenGL API 互換とする。

4.1. レンダリング手法の簡素化

本章の各手法は、各節に掲げる参考文献等に準じているが、以下のような簡素化を前提として高速化を図る。また今後のソフトウェア開発や不具合修正等の維持管理に係るコストを抑えるため、コーディングが徒に複雑とされないことも考慮する。

(a) 扱うプリミティブは三角形のみとする

各オブジェクトは、三角形の集合体として考える。小惑星の形状モデルはポリゴンモデルとして提供されており、曲面等を考慮する必要はない。探査機の形状等も、ポリゴンモデルに変換して使用する。

(b) 鏡面反射や透過は考慮しない

オブジェクトのレンダリングにあたって、鏡面反射や透過等は、運用者向けの可視化としては本質ではないため、割愛する。

(c) 小惑星や探査機を描く際の光源は太陽のみとする

陰 (shade) や影 (shadow) の計算にあたって、光源は1つのみとする。探査機のライトを用いた撮影等、複数光源の使用は特殊事例であり、現時点では考慮しない。

(d) 座標変換等はオブジェクト単位とする。

各プリミティブ毎の異なる座標変換は行わず、座標変換等が必要な場合はオブジェクトを単位とする。今後構築されるアプリケーションにて、探査機のアンテナ等、可動部分を取り入れる際は、稼働部分を独立したオブジェクトとして扱うことを想定する。

4.2. 形状を計算する手法

本稿においては、形状を計算する手法として、プリミティブ毎に計算する手法と画素毎に計算する手法を検証する。

4.2.1. プリミティブ毎に計算する手法

レンダリング対象のプリミティブ毎に、画面上の位置と色を計算する手法である。

本稿においては、計算するシーンの奥行き情報を参照する手法 (Z-Buffering^{*11}) を採用する。以後、奥行き情報を参照するバッファを深度バッファと記す。この深度バッファに記録された奥行き情報と新たに計算するプリミティブの奥行き情報を比較し、それまで計算した中で最短の奥行きとなる部分については、深度バッファに新たな奥行き情報を記録するとともに、その部分に表示されるべき色を、当該プリミティブと光源との位置関係等に基づいて計算する。

本稿においては、プラットフォーム間の互換性が高い高速化手段として、OpenGLとGPUの組み合わせを採用する。OpenGLには、`glVertexPointer()` や `glDrawArrays()` 等^{*12} の、予め定義された頂点配列等を一括して扱うためのAPIが用意されており、本稿においても、これらのAPIを用いて、オブジェクト単位でプリミティブの頂点配列を作成の上、レンダリングを行う。そのため、上記深度バッファに係る計算式は、ソースコード上には表れない。

陰やテクスチャ、歪みに係る将来の要求仕様に対しては、プログラマブルシェーダでの対応が想定されるが、現時点では固定機能シェーダ

^{*11} From Wikipedia, the free encyclopedia, “Z-buffering”, <https://en.wikipedia.org/wiki/Z-buffering> (2015年11月)

^{*12} Khronos Group, “OpenGL 2.1 Reference Pages”, <https://www.opengl.org/sdk/docs/man2/> (2015年11月現在)

を用いる。

この手順だけでは他のプリミティブと光源との位置関係を知る方法が無いので、影を計算することはできない。

4.2.2. 画素毎に計算する手法

レンダリング対象の画素毎に、その画素に表示されるべき色を計算する手法であり、本稿で述べる手法はレイトレーシング⁽³⁾の手法の一部である。

この手法においては、各画素の対応した視線(レイ)を追跡し、レイとプリミティブとの衝突位置を計算する。レイと衝突する全てのプリミティブに対して、その衝突位置までの距離を計算し、最も距離の近いものについて、その部分を構成するプリミティブと光源との位置関係等に基づいて陰を計算する。本稿においては、透過を考慮していないため、レイの追跡は1回のみである。

本稿においては、GPUを用いた高速化手法として、OpenCLを採用する。OpenCLは異なるアーキテクチャのGPUに対して同じソースを適用できるメリットがあると期待される^{*13}。

陰やテクスチャ、歪みに係る将来の要求仕様に対しては、関連するソースコードの追加で対応可能である。

プリミティブ毎に計算する手法と同様、この手順だけでは他のプリミティブと光源との位置関係を知る方法が無いので、影を表現することはできない。

4.3. 影を計算する手法

影を計算するには、光源と他のプリミティブとの位置関係を計算する必要がある。本稿においては前節4.2.1, 4.2.2の手法を光源方向に適用して計算する。また本稿においては、単一光源を前提としているため、いずれの手法における計算も単一光源に対して1式のみとなる。

4.3.1. プリミティブ毎に計算する手法

シャドウマッピング⁽⁶⁾の手法である。本稿においては、4.2.1との組み合わせをシャドウマッピングとして評価する。

4.2.1においては視点と各プリミティブとの位置関係から深度バッファを計算したため、その深度バッファは視点との距離情報を有する深度バッファとなる。本節においては、この方式を、光源と各プリミティブの位置関係に置き換えて計算する。すなわち光源を影計算の視点位置として、最も光源位置に近いプリミティブの深度を計算する。この深度バッファは光源との距離情報を有する深度バッファであり、シャドウマップと呼ばれる。

シャドウマッピングにおいては、視点との距離情報を有する深度バッファと、シャドウマップとを比較することで、それぞれの部分が最も光源に近いか、もしくは他のプリミティブにさえぎられて影になっているかを判定する。この手法においては、シャドウマップは常に影のレンダリング対象となるオブジェクトを包含するように設定する必要がある。

4.3.2. 画素毎に計算する手法

4.2.2と共にレイトレーシング⁽³⁾の手法になる。

各画素について視点とプリミティブとの最短距離が求まったところで、その距離に相当する位置を新たな視点位置として、4.2.2と同様の手法で光源方向のレイを追跡する。4.2.2においてはレイと衝突する全てのプリミティブとの距離計算が必要であったが、本稿においては影(shadow)の判定に透過を考慮していないため、レイがいずれかのプリミティブと衝突することが判明した時点で追跡を打ち切る。

また本稿においては、4.2.1との組み合わせも併せて評価する。4.2.1と4.3.2の組み合わせの名称については、該当する文献を見出すことができなかったため、仮に「ハイブリッド法」と称する。

5. 各手法の比較概略

前章で述べた手法について、精度や速度の観点での要約を表3に示す。表の中で、「○」は要求仕様を満たすための優位性があることを示す。「△」は条件次第で要求仕様を満たすことを示す。

^{*13} Khronos Group, “Conformant Products”,
<https://www.khronos.org/conformance/adopters/conformant-products#opencl> (2015年11月現在)

表 3 速度・精度面の比較

計算手法	プリミティブ毎	画素毎
形状の精度	△	○
影の精度	△ (*1)	○
GPU を用いた高速化	OpenGL	OpenCL
計算速度	○	△

(*1): 精度を要求されない用途での高速表示目的に限定される。

次に、評価対象として想定する手法の組み合わせと、本稿においてこれらの手法を区別するための名称を表 4 に示す。次章以降では、これらの手法について精度、速度の面でなされた検討の詳細を述べ、性能評価を行う。なお、形状を画素毎に計算し影をプリミティブ毎に計算する手法については、速度面でも精度面でも優位性が見いだせなかったため、本稿では割愛する。

表 4 評価対象として想定する手法の組み合わせ

形状の計算	影の計算	本稿での名称
プリミティブ毎	N/A	シェーディング (shading)
プリミティブ毎	プリミティブ毎	シャドウマッピング (shadow mapping)
プリミティブ毎	画素毎	ハイブリッド法 (hybrid method)
画素毎	プリミティブ毎	N/A
画素毎	画素毎	レイトレーシング (ray tracing)

6. 精度に係る検討

本章では、各計算手法の精度に係る課題を検討する。

レイトレーシングを OpenCL で実装する場合、32 ビット浮動小数点で計算するのが現実的である。32 ビット浮動小数点は 10 進数で 7 桁程度の精度を有している。これは 20km のレンジに対して 2mm 程度の精度であり、小惑星近傍のミッションを任意の視点から描いたとしても、要求仕様である 10cm の精度を満たす値である。

一方、プリミティブ毎に計算する手法では、深度計算に起因する精度不足や、深度バッファの解像度不足が生じる場合がある。以下に不具合の例を示す。位置・姿勢等のパラメータは表 5 に示す。その模式的な位置関係を図 1 に示す。この配置は、様々な不具合を一つのシーンに例示するための配置であり、現実の運用を反映したものではない。

表 5 不具合が生じる例の位置・姿勢等
探査機

位置	(0.06028, -0.10510, 0.06881)
姿勢	(0.80242; 0.42986, -0.35257, 0.21684)

小惑星

位置	(0, 0, 0)
姿勢	(0.13272; 0.65282, 0.73086, 0.14845)

太陽 レイトレーシング用 (点光源)

位置	$(1.04389 \times 10^8, 0.90274 \times 10^8, 0.37203 \times 10^8)$
----	---

太陽 シャドウマッピング用 (平行光源)

位置	(1.04389, 0.90274, 0.37203)
----	-----------------------------

方向	(-1.04389, -0.90274, -0.37203)
----	--------------------------------

カメラ

位置	(0.11504, 0.07638, 5.00000)
----	-----------------------------

方向	(0,0,1)
----	---------

視野角	0.23097 度
-----	-----------



図 1 模式的な位置関係

位置の実際の単位は [km] であるが、本稿においては 1km がプログラム上の表現で 1 となるように各表示すべきオブジェクト等のスケールを合わせている。姿勢は SPICE Toolkit^{*14} が提供するクォータニオンである。太陽は点光源もし

^{*14}Navigation and Ancillary Information Facility, NASA, “SPICE Conventions”, https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/05_conventions.pdf, p.13 (2015 年 11 月現在)

くは平行光源として扱い、白色光とする。影の計算にあたって、画素毎に計算する手法では点光源の値を用い、プリミティブ毎に計算する手法では、平行光源の値を用いる。カメラの視野角は、現実のカメラでは想定し難い値に設定しているが、GUI操作により自由に拡大・縮小するような運用においては起こりうる値であり、可視化手法の中でも考慮しておく必要がある。

この配置に基づいたレイトレーシングによるレンダリング例を図2に、シャドウマッピングによるレンダリング例を図3に示す。

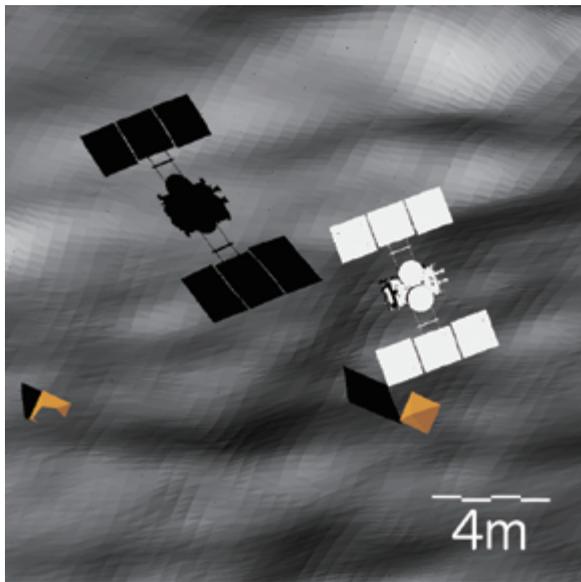


図2 レイトレーシングのレンダリング例

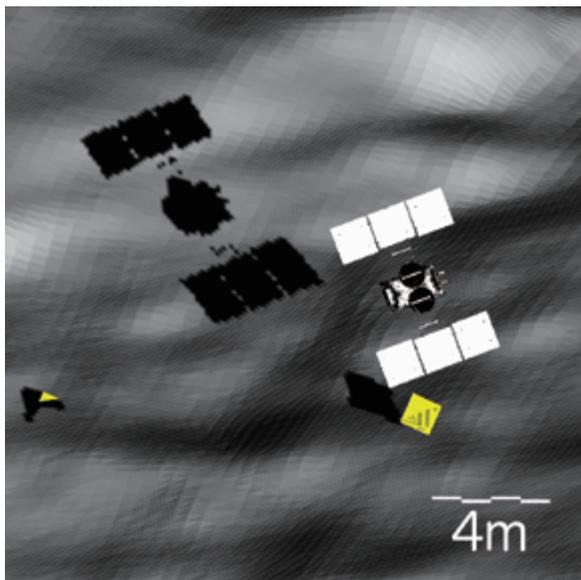


図3 シャドウマッピングのレンダリング例

各図において、中央右付近のモデルが探査機、背景の灰色の多面体が小惑星を表している。四角錐はLRFによる測距値に基づいて計算した小惑星面の推定位置を表しており、図中には2つ描かれている。右下の「4m」の表示は、探査機の位置での大まかな4mの長さを表している。横棒1本は1mを表している。

これらの図から読み取れる問題点は、次の通りである。

6.1.1. 深度バッファの精度不足

OpenGLにおける深度バッファは、奥行きを最小値と最大値を設定する必要がある。レンダリングされるのは、この設定範囲内のオブジェクトのみである。また透視投影を表現するためのパースペクティブ行列と深度バッファを併用する場合、深度バッファは近距離において高精度となり遠距離においては低精度となる。環境によっては深度バッファが32ビットではなく、24ビットの精度しか得られない場合もあり^{*15}、注意が必要である。

このような条件下においては、直近のオブジェクトと遠方のオブジェクトを同時に描くことが困難な場合がある。深度バッファの精度に関する課題は、この例に限らず、深度バッファとパースペクティブ行列を併用する場合に共通に発生しうる。

探査機と小惑星のランデブーについて、概ね20km程度の距離からタッチダウンまで描くと想定すると、視点や注視点としては1mオーダから10kmオーダの範囲でオブジェクトを描くことになると考えられる。単純にこの想定からOpenGLの深度範囲を1m～10km相当に設定して描いたものが図3である。

図3は、探査機の形状が崩れており、また周囲のLRFに係る四角錐も不適切なレンダリングになっている。不具合の拡大例を図4に示す。各図共に、レイトレーシング側の画像は、適切に形状を反映している。

^{*15} Apple Inc., “OpenGL Capabilities Tables”, <https://developer.apple.com/opengl/capabilities/> (2015年11月現在)

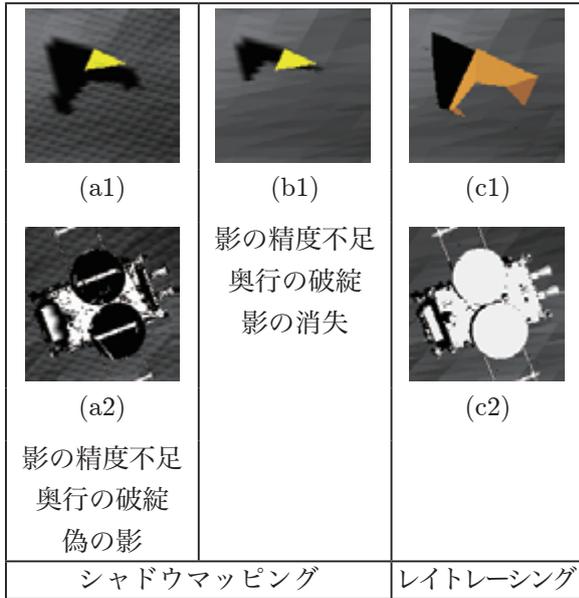


図 4 精度に係る不具合例 (拡大)

図 4(a1) においては、LRF に係る四角錐と小惑星との奥行関係が破綻して、本来描かれるべき錐の一部が消えている。この例では、四角錐のサイズは 1 辺が 1m 程度であるので、そのサイズに匹敵する部分が消失しうることが見て取れる。図 4(a2) においては、探査機内のプリミティブ同士の奥行関係が破綻して、本来奥に隠れているはずの構造が露呈している。

このような問題を回避するためには、探査機や小惑星等、描くオブジェクトと視点との位置関係を常に把握しながら、レンダリングするシーンの奥行き範囲を適切に決定する必要がある。また場合によっては、精度低下が懸念されるような視点やオブジェクトの配置を禁止するような工夫も必要になると考えられる。この留意点は表 4 に掲げたシェーディング、シャドウマッピング、ハイブリッド法に共通するものである。

6.1.2. シャドウマップの精度不足

シャドウマップに起因する課題も見受けられる。シャドウマップも深度バッファであるため、6.1.1 で述べた課題が発生しうが、これはシャドウマップを計算する際に並行投影 (並行光線) を用いることで軽減しう。その他の課題を以下に述べる。

例えば本稿で用いたシステムにおいては、シャドウマップを計算するための深度バッファのサイズは、4096×4096 ピクセルが限界であった。「は

やぶさ 2」が目的としている小惑星 1999 JU₃ は、直径が 1km 弱と推定されており、これで小惑星の存在範囲をカバーするとすると、深度バッファ 1 ピクセル当たりのサイズは 25cm 四方程度となり、精査目的では要求仕様を満たさない。

また本来はありえないテクスチャが図 3 の随所に見受けられる。図 4(a1), (a2) において、小惑星上の細かい模様は、シャドウマップの量子化誤差に起因する偽の影であり、実際には存在しないものである。本稿においてはテクスチャを単色でレンダリングしているため偽の影を見分けることが容易であるが、将来的に小惑星のテクスチャを付加した画像においては、それが本来のテクスチャであるのか量子化誤差に起因するのを見分けるのは困難となる。

このような量子化誤差の影響を軽減するためには、シャドウマップに若干の不感域を設けて、影の判定が際どい領域では影と判定しない等の対策が必要となる。しかしながらこのような対策を講じると、逆に本来は影であるべき領域が影と判定されなくなる副作用がある。図 4(b1) は、小惑星上の偽の影が目立たなくなる程度に不感域を設けた場合の例である。LRF に係る四角錐の影は、図 4(c1) のように描かれるべきところであるが、その多くの部分が消失している。

シャドウマッピングを用いて影を描く際には、これらの誤差を考慮しながら、影が適切に描かれるようパラメータを調整しなくてはならないが、実際問題として、運用者がこの調整を行うことは現実的ではない。

このように、シャドウマッピングは影の精度が要求される場面では利用が困難である。

7. 高速化に係る検討

OpenGL を用いてプリミティブ毎に計算する手法であれば、GPU による高速化が容易であり、速度に関する懸念は小さい。シャドウマッピングも、画像を精査する必要の無い場面では、速度面の優位性がある。またシャドウマッピングはソースコードが比較的シンプルであり、保守のメリットも勘案しつつ、精度を要求しない場面での簡易かつ高速表示の手段として候補に残すこととする。

これに対して画素毎に計算する手法は、それ

それぞれの画素に対して各プリミティブとの衝突判定を行うため、計算時間を要する。以下に、画素毎に計算する手法の高速化について検討する。

7.1. ボクセル分割

一般のレイトレーシングと同様に、4.2.2と4.3.2で示した、画素毎に計算する手法においては、ボクセル分割⁽⁷⁾による高速化を行う。本稿においては、下位層のボクセルは上位層のボクセルを各軸方向に二等分する⁽⁸⁾方式を採用する。すなわち、上位層のボクセルから、夫々8個の下位ボクセルを生成する、8分木の構成となる。

ボクセル分割は、シーンの空間に沿って分割して階層化する方法やオブジェクトに沿って分割して階層化する方法等が考えられる。シーンの空間に沿って分割する場合、シーンのレンダリング毎にボクセル分割する必要があるが、ボクセルの階層化や、階層化されたボクセルにプリミティブを割り当てる作業はOpenCLによって並列度を上げることが困難である。

本稿で扱うオブジェクトについて考えると、個々のオブジェクトの座標系において、それに含まれるプリミティブが変形することは考慮しなくて良い。個々のオブジェクトに沿ってボクセル分割する場合、階層化処理はオブジェクト毎に1度実施すれば、一連のシーンのレンダリングに渡って、そのオブジェクトのボクセル構成は不変であるとみなすことができる。また本稿で想定するシーンは、プリミティブ数は多いが、オブジェクト数は少ないため、オブジェクト毎にボクセル階層を持つオーバーヘッドは小さいと考えられる。

そこで本稿においては、オブジェクトに沿って8分木に階層化する方法を採用する。階層化されたボクセルの内、最下層のボクセルには、それぞれに（全体もしくは一部が）内包されるプリミティブを割り当てる。8分木による階層化の例を図5に示す。

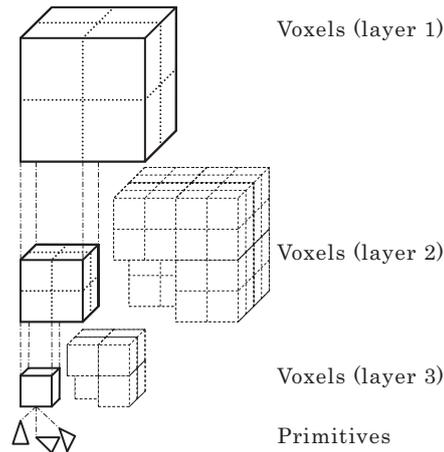


図5 ボクセル分割の概要

本稿で採用するボクセルの階層構造においてプリミティブとレイとの衝突位置を計算するにあたっては、最初に計算対象となるレイの始点とレイの方向を、当該オブジェクトの座標系に変換する。各オブジェクトの座標系内において、最上位のボクセルからレイの衝突判定を行い、レイと衝突したボクセルについては、下位層のボクセルとの衝突判定を順次実施する。衝突判定された最下層のボクセルについては、そこに含まれるプリミティブに対してレイの衝突判定を行う。この手順を、表示すべき全オブジェクトに対して実施する。

本稿で用いたボクセル探索の概要を図6に示す。

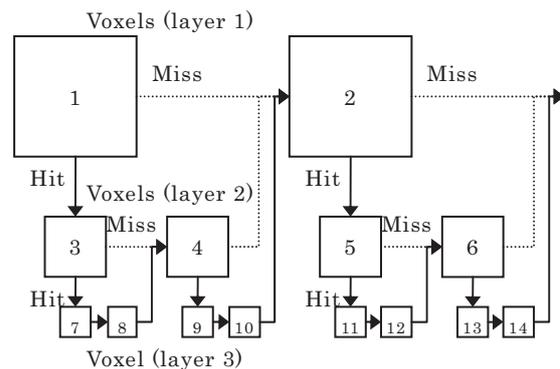


図6 ボクセル探索の概要

図中では簡単のため2分木を3階層で表現しているが、実際の探索には8分木が用いられ、階層数もオブジェクトによって異なる。

またOpenCLにおいては、ホスト側のポインタ変数をGPUでは参照できないため、ボクセルの探索には、インデックスによる間接参照を採

用した。図 6 の探索をインデックスで表現した例を表 6 に示す。

表 6 ボクセル探索のインデックス表現の例

Voxel #	1	2	3	4	5	6	7	8
Next voxel # (hit)	3	5	7	9	11	13	8	4
Next voxel # (miss)	2	...	4	2	6	...	8	4

7.2. GPU による高速化の範囲

4.2.1 と 4.3.1 で述べた、プリミティブ毎に計算する手法については、OpenGL と GPU の組み合わせで高速化する。4.2.2 と 4.3.2 で述べた画素毎に計算する手法については、OpenCL と GPU の組み合わせで高速化する。

GPU と OpenGL や OpenCL の互換性については、OpenGL, OpenCL 共に、3.1 に記した AMD 社, NVIDIA 社, Intel 社の各 GPU について動作検証を行う。

評価対象となる手法の組み合わせと、本稿で用いる各組み合わせの名称は、前述の表 4 に示した通りであるが、この中でハイブリッド法は、OpenGL と OpenCL という異なる手段でレンダリングを分担する手法となっている。すなわち、OpenGL で形状を計算した結果である画像の配列と深度バッファの配列を OpenCL に引き継いで影の計算を実施する。この引き継ぎ過程が入るため、GPU での計算時間が短い場合は引き継ぎによるオーバーヘッドが相対的に大きくなると考えられる。

OpenGL での処理と OpenCL での処理とで座標計算を一致させる必要もあるが、本稿においては座標計算用の行列を両者互換となるよう設計しているため、座標計算に関しては特にオーバーヘッドは発生しない。

8. 所要時間等の比較

8.1. 比較対象

所要時間の比較は、表 4 に掲げた手法に対して実施する。計測にあたっては原則として GPU による高速化を行うが、レイトレーシングについては、CPU による所要時間も併せて評価する。これは GPU 搭載のメモリでは処理できないよう

な大規模形状データのプリレンダリングも模擬・訓練の中で想定しうるためである。CPU によるレイトレーシングについては、Grand Central Dispatch (GCD)^{*16} によるマルチスレッド化を行う。GCD は本稿で用いたプラットフォームに限定されるものではないが、他の OS で係るアプリケーション開発を行う場合は、その互換性に留意する必要がある。またプリレンダリングにおいてはパラメータ並列の計算も可能であり、レンダリングする一連のシーン全体としての高速化にマルチスレッド化が必須とは限らない。

8.2. 比較に用いるシーン

比較に際して表示すべきオブジェクト等の配置を表 7 に示す。各値の定義等は、表 5 と同一である。これらから描かれるシーンの例を図 7, 図 8 に示す。シーンのサイズは、1,024×1,024 とする。

表 7 表示すべきオブジェクト等の位置・姿勢
探査機 各シーン共通

位置	(0.04061,0.13302,0.07361)
姿勢	(0.78839; 0.51098, -0.30285, 0.16005)
小惑星 各シーン共通, 4 モデル共通	
位置	(0, 0, 0)
姿勢	(0.10212; 0.50216, 0.84153,0.17096)
太陽 点光源	
位置	(1.04455×10 ⁸ ,0.90215×10 ⁸ ,0.37176×10 ⁸)
太陽 平行光源	
位置	(1.04455, 0.90215, 0.37176)
方向	(-1.04455, -0.90215, -0.37176)
カメラ a シーン a 用	
位置	(0.04306, 0.05716, 5.00000)
方向	(0,0,1)
視野角	0.41815 度
カメラ b シーン b 用	
位置	(0.00977, 0.03872, -5.00000)
方向	(0,0,1)
視野角	4.93447 度

^{*16} Apple Inc., “Grand Central Dispatch (GCD) Reference”, https://developer.apple.com/library/mac/documentation/Performance/Reference/GCD_libdispatch_Ref/index.html (2015 年 11 月現在)

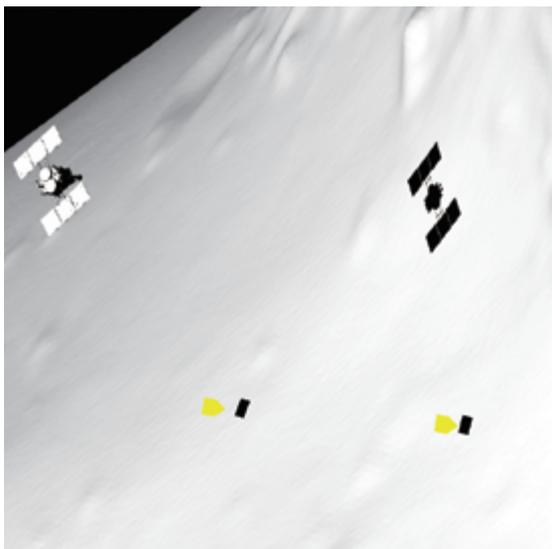


図 7 所要時間比較に用いるシーン a (近景)

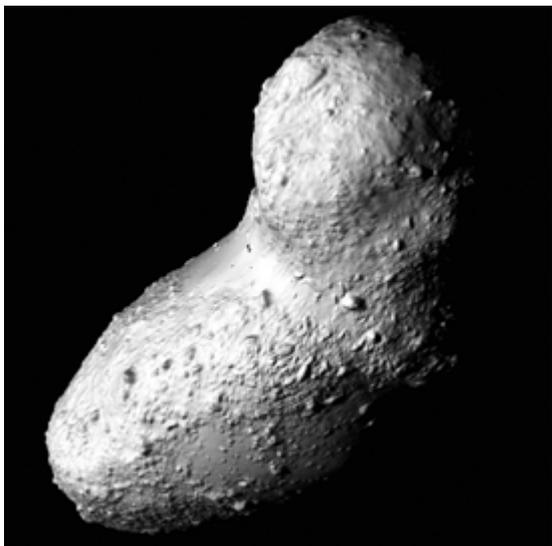


図 8 所要時間比較に用いるシーン b (遠景)

8.3. ボクセル分割の階層数

画素毎に計算する手法で用いたボクセル分割の階層数やボクセル数を、表 8 に示す。階層数は、現在のところ経験則で高速となる値を求めており、最適な階層数を自動的に求める手法については、今後の課題である。

なお Mac mini と 3,145,728 ポリゴンモデルの組み合わせにおいては、OpenCL を用いたレンダリングで GPU のメモリ容量の限界を超えたため、OpenCL を用いたレイトレーシング及びハイブリッド法の階層数を 1 つ減じている。ケース 2 については GPU のメモリ容量を超える事例がさらに多く見受けられたため、OpenCL を用いる手法については GPU で正常に処理できるも

のみ計測した。GPU のメモリ容量を勘案した階層数の設定も、今後の課題である。

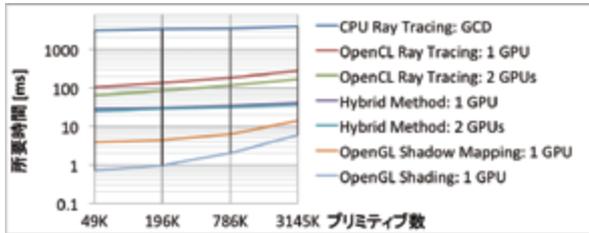
表 8 ボクセル分割の階層数・ボクセル数

モデル	プリミティブ数	ケース 1		ケース 2	
		階層	ボクセル	階層	ボクセル
小惑星	49,152	8	37,874	9	150,515
	196,608	9	151,186	10	600,321
	786,432	10	601,285	11	2,384,543
	3,145,728	11	2,387,458	12	9,460,613
探査機	213,405	9	114,047	10	577,835

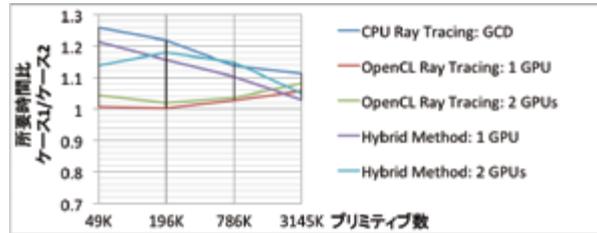
8.4. 比較結果

各手法においてレンダリングに要する時間の比較結果を図 9、図 10 に示す。横軸は小惑星のポリゴン数、縦軸はシーン 1 回のレンダリング時間を表している。小惑星のポリゴン数は、1 段詳細化されるごとに 4 倍のプリミティブ数となっているため、事実上横軸はログスケールとなっている。

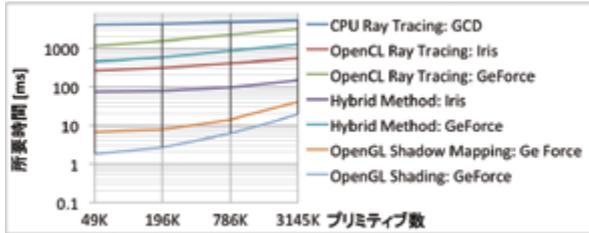
レンダリング時間の計測にあたっては、各条件ごとに同一のレンダリング計算を 100 回連続して実施し、その平均をとった。ケース 2 で正常にレンダリングできた事例については、図 11、図 12 に、ケース 1 との所要時間比を示す。



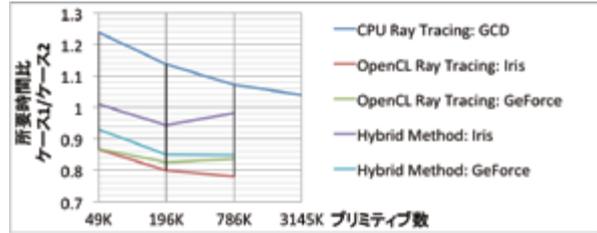
MacPro



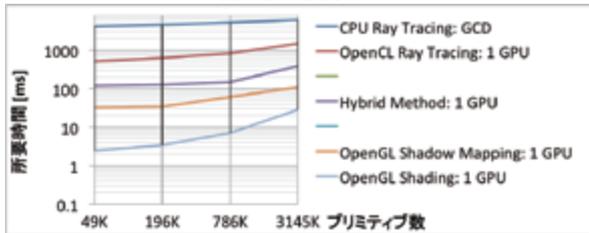
Mac Pro



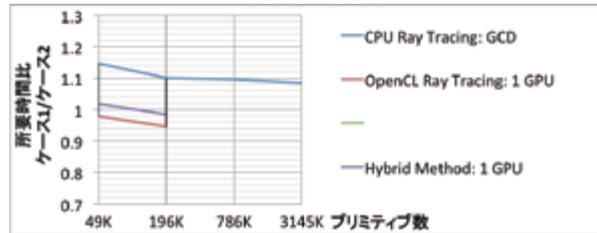
MacBook Pro



MacBook Pro



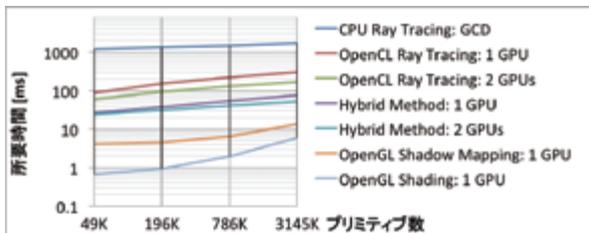
Mac mini



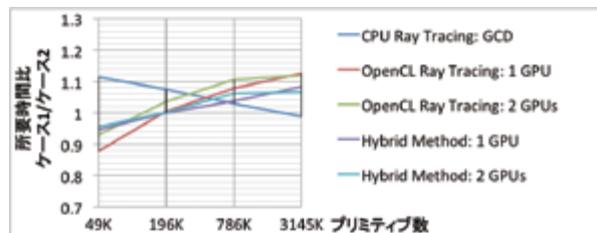
Mac mini

図 9 レンダリング速度比較 (ケース 1, シーン a)

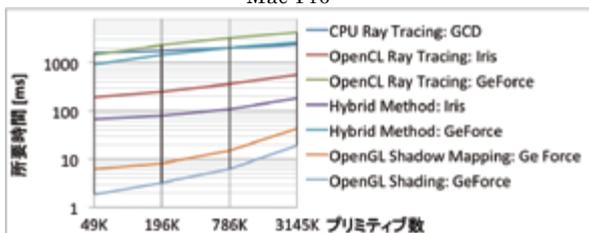
図 11 ボクセル階層数による所要時間比較(シーン a)



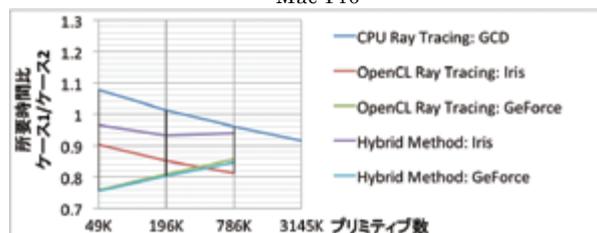
Mac Pro



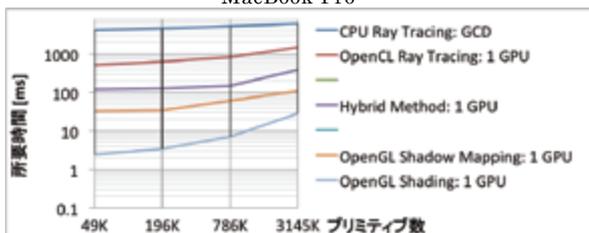
Mac Pro



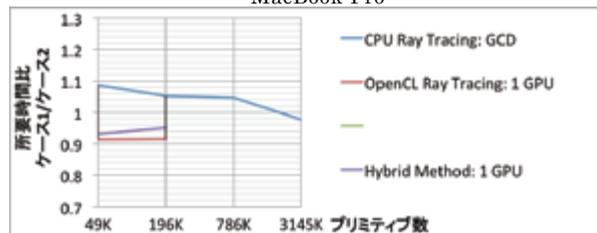
MacBook Pro



MacBook Pro



Mac mini



Mac mini

図 10 レンダリング速度比較(ケース 1, シーン b)

図 12 ボクセル階層数による所要時間比較(シーン b)

いずれのケースが高速にレンダリングできているかについては、実行条件によって様々であり、一概にいずれの階層数が優れていると結論付けるのは難しい。階層数等、ボクセル分割に係るチューニングは、今後ソフトウェアを開発するプラットフォームが決定した後の検討事項になると考えられる。

次に測定環境毎の結果を述べる。

(a)Mac Pro 環境

Mac Pro は同一型の GPU を 2 基搭載しているため、OpenCL を用いた高速化の評価にあたっては、1GPU 単独の評価と 2 GPU 並列の評価を行った。

本稿における評価の中で最もポリゴン数の多い 3,145,728 プリミティブの場合でも、1GPU の場合、ハイブリッド法が高速表示の要件 (10fps) を満たしており、OpenCL を用いたレイトレーシングもかろうじて精細表示の要件 (3fps) を満たしている。ハイブリッド法において、深度バッファに係るパラメータ設定を適切に行えば、あらゆる場面で高精細かつ高速の表示が実現する可能性がある。2GPU のレイトレーシングは余裕を以て精細表示の要件を満たしている。

1GPU と 2GPU との計測結果を比較し、2GPU 時の性能向上比を求めたものを図 13 に示す。レイトレーシングにおいては、性能向上の効果が大きい。ハイブリッド法においては、OpenCL でのレンダリングが 1GPU に限定されることと OpenCL・OpenCL 間のデータ引き継ぎのオーバーヘッドもあって、性能向上の効果は低い。

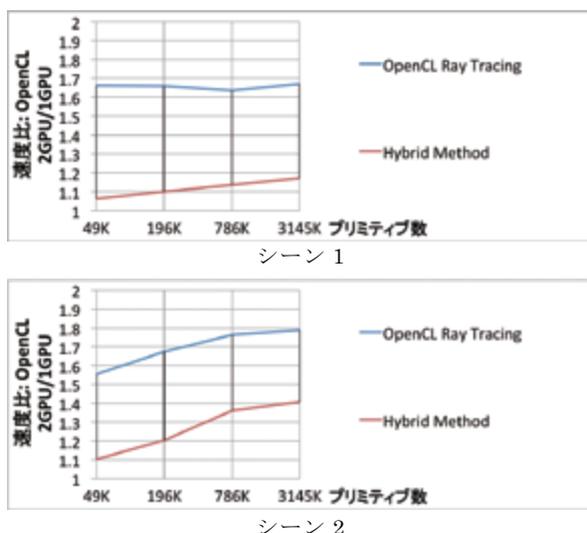


図 13 1GPU と 2GPU での速度比

CPU を用いたレイトレーシングにおいては、1 秒～数秒で 1 枚のレンダリングができており、大規模形状モデルを用いたプリレンダリングにおいては、64GB のメモリを有効に活用できるものと考えられる。

(b)MacBook Pro 環境

MacBook Pro 環境においては、OS の制約により、ディスプレイ表示用 (OpenGL 実行用) の GPU は自動的に GeForce が割り当てられた。図 9 から図 12 におけるハイブリッド法の GPU は、OpenCL を実行した GPU を表している。

ここでは Iris Pro を用いたハイブリッド法が精細表示の要件を満たしている。速度面ではシャドウマッピングもしくは Iris Pro を用いた低ポリゴン数のハイブリッド法が高速表示の要件を満たしている。このようなモバイル環境でのプレゼンテーション等においては、シャドウマッピングとハイブリッド法を適切に使い分けることにより、実用的な表示が実現できる可能性がある。

MacBook Pro と、Iris Pro の低速版にあたる GPU を搭載した Mac mini とで、シェーディングの所要時間差は 10ms 程度である。MacBook Pro にて OpenGL を含めて Iris Pro で実行したとしても、本稿で計測した所要時間と比較して、その差は僅かであると想像される。これは GPU 内蔵の CPU パッケージでリアルタイムレンダリングが実用化されうること示唆している。

GeForce を用いた OpenCL 関連の所要時間が大きくなっているが、これは OpenCL と GeForce のアーキテクチャの不整合によるものなのか、ディスプレイ表示と OpenCL を兼用したことによるものなのかについて、精査する余地がある。MacBook Pro の OS には、ディスプレイ表示用の GPU を手動で切り替えるような環境設定が用意されていないため、原因切り分けのためには、別途 GeForce を OpenCL 専用にした場合の性能評価を行う必要があると考えられる。また NVIDIA 社のアーキテクチャ専用の CUDA^{*17} による評価も選択肢の一つであろう。

(c)Mac mini 環境

Mac mini 環境の性能やメモリ割当では実利用

^{*17} NVIDIA Corporation, "Parallel Programming and Computing Platform | CUDA | NVIDIA | NVIDIA", http://www.nvidia.com/object/cuda_home_new.html (2015 年 11 月現在)

に供することは困難と考えられるが、開発時の機能確認や互換性確認の用途には利用可能であると考えられる。

9. おわりに

探査機運用者向けの可視化手法における、影の正確な描写と応答速度の向上について検討した。

高精度の描写に長けているレイトレーシングについては、インタラクティブなレンダリングにおいても、環境によっては実用的な応答時間で利用可能であることが示唆された。本稿においては、そのために不要な計算要素の省略や、レイの追跡アルゴリズムの簡略化を検討し、また OpenCL による高速化を施した。

ハイブリッド法は、本稿で述べた運用者向け可視化の範囲においては、深度バッファのパラメータを考慮することで精度を維持しつつ、レスポンスの向上も期待でき、一層の調査・研究が有望であると考えられる。

今後はアルゴリズム及び GPU 選定の双方から調査・検討を行い、高精度な影の描写をリアルタイムに実現できるように改善を図って行く予定である。

また執筆時点で用いているレイトレーシングやハイブリッド法のアルゴリズムは、係る行列計算等を忠実にコーディングしたものであり、GPU の特性を考慮したチューニングには至っていない。可視化ソフトウェアを搭載する機種が選定された後は、GPU の特性を考慮したチューニングを行うことでさらなる速度向上が期待できる。

また本稿で評価に用いた GPU は、最速のもの (FirePro D700) でも、執筆時点で既に販売開始から 2 年経過した製品であり、執筆時点の性能としては、この 2 倍以上の演算性能を実現した製品が見受けられる^{*18,*19}。ソフトウェアの改良のみならず、最新のハードウェアを用いることによって、2018 年の運用時点では、さらなる速度向上が見込めるものと期待される。

また、本稿の評価の中には、ディスプレイ表示に割り当てられている GPU を OpenCL に供した場合の評価も含まれている。その場合、速度評価のプログラム実行中は他のウィンドウ操作や文字入力等に遅延が見受けられることがあった。本稿のような機能・性能を評価する段階では他の操作を控えての実行が可能であるが、実運用環境で GUI 操作をする場合、操作に係る遅延も考慮する必要がある。速度向上と操作性の向上とのトレードオフは今後の課題である。

参考文献

1. 「はやぶさ」の小惑星イトカワ探査軌道・姿勢の可視化. 三浦昭, 山本幸生, 吉川真. 2014 年, 宇宙科学情報解析論文誌 第 3 号, ページ: 7-15.
2. はやぶさの軌跡の可視化: タッチダウン時の位置推定. 三浦昭, 山本幸生, 吉川真. 4, 2015 年, 宇宙科学情報解析論文誌 第 4 号, ページ: 173-183.
3. Some techniques for shading machine renderings of solids. Appel, Arthur. 1968, AFIPS '68 (Spring) Proceedings of the April 30--May 2, 1968, spring joint computer conference, pp. 37-45.
4. Bidirectional Reflectance Spectroscopy 1. Theory. Hapke, Bruce. B4, Journal of Geophysical Research, Vol. 86, pp. 3039-3054.
5. 再び宇宙大航海へ臨む「はやぶさ 2」第 5 回航法誘導制御. 照井冬人. 399, 2014 年, ISAS ニュース, ページ: 5.
6. Casting Curved Shadows on Curved Surfaces. Williams, Lance. 1978, SIGGRAPH '78 Proceedings of the 5th annual conference on Computer graphics and interactive techniques, pp. 270-274.
7. Reddy, Dabbala Rajagopal and Rubin, S. Representation of three-dimensional objects. s.l. : Department of Computer Science, Carnegie Mellon University, 1978. pp. 1-16.

^{*18} From Wikipedia, the free encyclopedia, "GeForce 900 series", https://en.wikipedia.org/wiki/GeForce_900_series (2015 年 11 月現在)

^{*19} From Wikipedia, the free encyclopedia, "AMD Radeon Rx 300 series", https://en.wikipedia.org/wiki/AMD_Radeon_Rx_300_series (2015 年 11 月現在)

8. Space Subdivision for Fast Ray Tracing.
Glassner, Andrew S. 1984, Computer Graphics and Applications, IEEE, pp. 15-24.
9. Shadow Algorithms for Computer Graphics.
Crow, Franklin C. 1977, SIGGRAPH '77 Proceedings of the 4th annual conference on Computer graphics and interactive techniques, pp. 242-248.

JAXA OPEN API による地球観測データの利用拡大

秋山 恭平^{*1}, 可知 美佐子^{*1}, 西原 雄次^{*1}, 館下 博昭^{*1}, 安部 眞史^{*1},
高井 元^{*1}, 石井 宏宗^{*1}

Expanding the Use of Earth Science Data via JAXA OPEN API

Kyohei Akiyama^{*1}, Misako kachi^{*1}, Yuji Nishihara^{*1}, Hiroaki Tateshita^{*1}, Masashi Abe^{*1},
Moto Takai^{*1}, Hiromune Ishii^{*1}

Abstract

JAXA OPEN API is a Web API (Application Programming Interface) designed to provide simple and efficient access to people beyond the public sector, including scientific and educational fields, with a view to promoting the broader sharing of Earth science data. In order to demonstrate the efficacy and usability of the API, Japan Aerospace Exploration Agency (JAXA) held a competition to develop innovative applications by taking advantage of the global Earth science data. This paper describes the activity and results for expanding the use of Earth science data using the JAXA OPEN API.

keywords: Web API, Earth Science data, application, competition

概要

宇宙航空研究開発機構 (JAXA) が所有する膨大な地球観測データを、異なる分野のユーザであっても簡単に利用できる仕組みを提供し、データの利活用用途を広げることを目的として、地球観測データの WEB API (JAXA OPEN API) を整備した。また、その API を利用したアプリケーションの開発コンテスト (JAXA OPEN API COMPETITION) を開催し、地球観測データを利用した多くのアプリケーションが創造されることを実証した。本稿では、これらの一連の活動と本活動がもたらす成果について述べる。

1. はじめに

スマートフォンを始めとする Information and Communication Technology (ICT) 技術の発達や Twitter や Facebook のような Social Networking Service (SNS) の普及により、小規模なコミュニティもしくは個人が、社会を変えるようなサービスや価値を生み出すことが可能になってきた。また、データ解析技術の発達により、その道の専門家でなくとも Web 上の大容量のデータ (ビッグデータ) を扱えるようになり、データの新しい利用価値を生み出す基盤が整えられつつある。さらに近年では、二次利用の可能なルールで公開された公共データ (オープンデータ) の利用促進を目的としたさまざまな取り組みが、国を挙げて進められている。

一方で、宇宙データは歴史も古く、ビッグデータの先駆けともいえるデータであるが、その利用は

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

^{*1} 宇宙航空研究開発機構 (JAXA)

公的機関（大学、独法、官公庁、自治体）を中心として、一部の専門家、研究機関、企業等がほとんどであり、一般への普及が進んでおらず認知度も低いのが現状である。しかし、上述のような技術革新の恩恵をうけ、ビッグデータである宇宙データを専門家以外の一個人が扱い、新たな価値を生み出すことが現実になりつつある。このような社会情勢の中で、NASA が企画する”International Space Apps Challenge (ISAC)^{*2)}” など、宇宙関連のオープンデータの新たな利用方法を模索する動きが世界各地で始まっている。

こうした現況を踏まえ、宇宙航空研究開発機構（JAXA）では、JAXA が所有する膨大な地球観測データを、これまでそのようなデータに触れたことのない他分野のユーザが利用できる仕組みを提供するとともに、アプリケーションを開発するきっかけとなる機会を提供することで、オープンデータの新たな利活用分野を創出する取り組みを始めた。具体的には、JAXA の地球観測データの WEB API（Application Programming Interface）を公開するとともに、本システムで提供されるデータを使ったスマートフォンや WEB サービスなどのアプリケーションの開発コンテストを開催した。

本稿では、JAXA の地球観測データの WEB API（JAXA OPEN API）のシステム概要を説明するとともに、この API を利用したアプリケーションの開発コンテスト（JAXA OPEN API COMPETITION^{*3)}）の成果について述べる。

2. JAXA OPEN API

2.1. システム概要

JAXA OPEN API は、宇宙データを「シンプルで直感的な形で」ユーザに提供することを目的としたウェブサービス（WEB API）であり、2013年11月より試行運用を開始した。本システムでは、地球観測データを、「物理量の種類+緯度・経度・日付」という直観的にイメージしやすい条件で指定することにより、必要な情報に必要な最小限の知識でアクセスすることができるように配慮されている。また、スマートフォンや WEB アプリでの利用を想定し、HTTPS プロトコルの GET メソッドでデータをリクエストすることで、リクエストパラメータに基づいて機械判読に適したデータ形式（XML、JSON 等）でデータ出力する機能を備えている。本システムが提供する地球観測データの利用ポリシーは、源泉となる地球観測データの利用規約（GCOM-W1 データ提供サービス^{*4)}、全球降水マップ GSMaP^{*5)}）に準拠している。また、これらのデータ提供サービスでは事前のユーザ登録を必要としていることから、本システムにおいても、利用者は事前にユーザ登録を行い、アクセスには認証用のトークンを必要とする設計としている。

本システムでは、ウェブサービスの実装を支援する Java ベースの Jersey をフレームワークとして採用し、そのフレームワーク上に Web API のプログラム（Java）が実装されている。また、トークン認証やデータの検索、クライアントへのレスポンスを行う個別プログラムが動作するサーブレットコンテナ（Tomcat）、クライアントからのリクエストを処理する HTTP サーバ（Nginx）、そして地球観測データおよびトークン認証用のトークンが格納されたデータベース（PostgreSQL）から構成される。JAXA OPEN API の全体システムの構成を図 1 に示す。

^{*2)} International Space Apps Challenge (ISAC), <https://spaceappschallenge.org>

^{*3)} JAXA OPEN API COMPETITION, http://www.satnavi.jaxa.jp/jaxa_api_competition

^{*4)} GCOM-W1 データ提供サービス, <https://gcom-w1.jaxa.jp>

^{*5)} 世界の雨分布速報, <http://sharaku.eorc.jaxa.jp/GSMaP>

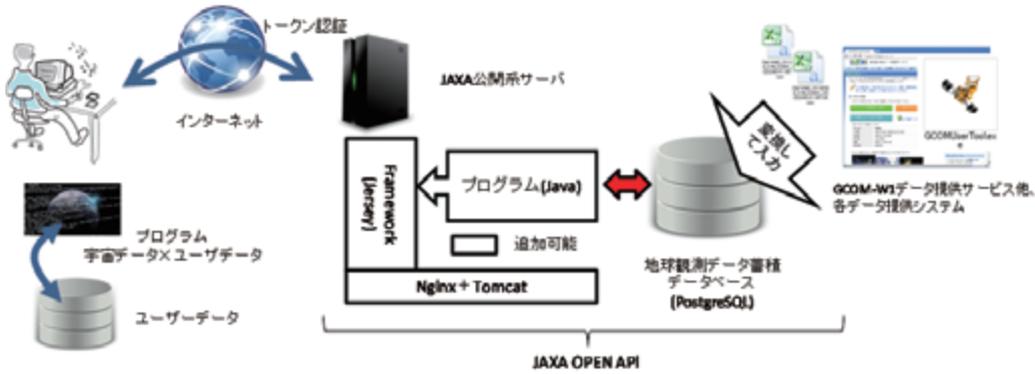


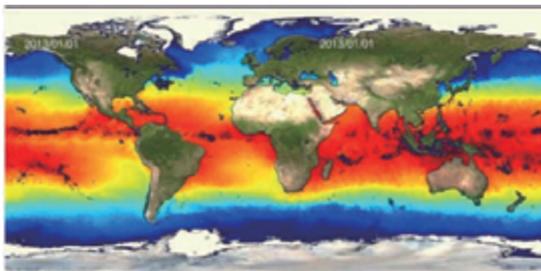
図 1 JAXA OPEN API のシステム構成

2.2. 提供データ概要

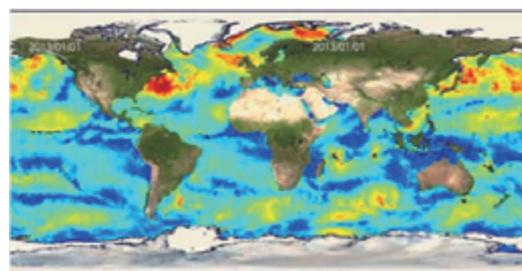
2015年2月現在、JAXA OPEN APIが提供する地球観測データは、①海面水温、②海上風速、③土壌水分量、④積雪深、⑤降水量の5つの物理量であり、2012年8月1日から現在（最新データを2日遅れで更新）までのデータを公開している。これらの地球観測データは、もともと以下に示す2つのデータ提供サービスからバイナリ形式のフォーマットで提供されているものを源泉としている。JAXA OPEN APIでは、これらのサイトから提供されているファイルから必要なデータ（地球物理量、緯度経度情報、時刻情報等）を抽出し、必要な空間分解能、時間分解能でシステム内のデータベース（PostgreSQL）に蓄積しており、ユーザから指定された条件に基づいて検索し、データタイプに応じた提供を行っている。それぞれの物理量プロダクトのイメージ及びデータ提供サービスの概要を以下に示す。なお、JAXA OPEN APIで取得できるデータの詳細は参考文献(1)を参照されたい。

GCOM-W1 データ提供サービス*4

第一期水循環変動観測衛星「しずく」(GCOM-W1)に搭載されているマイクロ波放射計(AMSR2)が観測した大気中の水蒸気量や海面の温度などの物理量である。マイクロ波は水に感度をもち、マイクロ波放射計は「水の情報」を持ったマイクロ波エネルギーを計測する。地表面や大気中の物体を放射源とするために太陽光を必要とせず、夜間や冬季極域の観測も得意とする。図2には、JAXA OPEN APIで提供されているGCOM-W1地球物理量プロダクトの例を示している。このうち、(a)海面水温と(b)海上風速は全球洋上を、(c)土壌水分量と(d)積雪深は陸圏を対象領域としている。



(a) 海面水温



(b) 海上風速

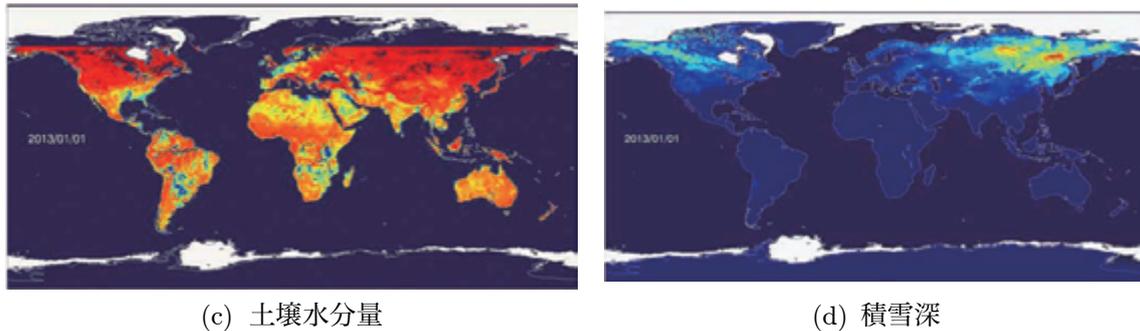


図 2 GCOM-W1 地球物理量プロダクトの例

衛星全球降水マップ GSMaP^{*5}

衛星全球降水マップ (GSMaP) は、複数の衛星の観測データを利用することで、世界の雨分布を準リアルタイム (観測から約4時間遅れ) で1時間ごとに提供している。図3には、JAXA OPEN API で提供されている GSMaP プロダクトの例を示している。北緯60度～南緯60度の範囲を対象領域としている。

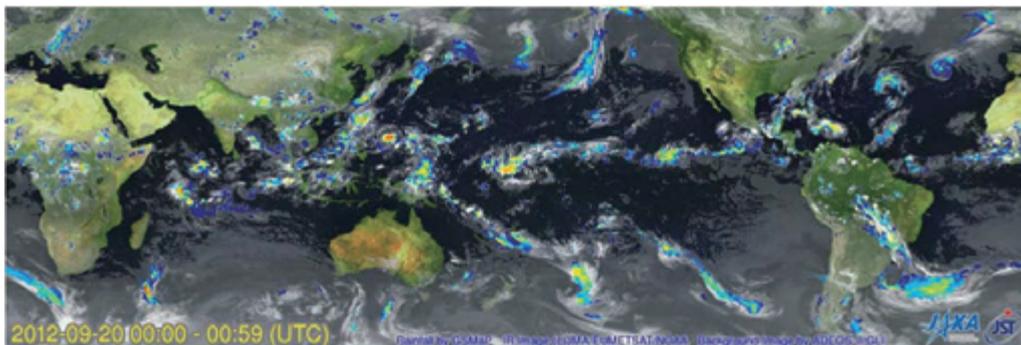


図 3 GSMaP プロダクトの例 (2012年9月20日の降水量)

2.3. データ解像度

Web サーバでのデータ取得依頼受付により、ユーザの取得要求に基づいてデータベース上からデータ検索を実施する。具体的には、特定の緯度・経度・日付等の取得要求に対して、要求された緯度・経度領域を含む観測点をデータベース上から検索し取得する。ただし、地球観測データの場合、データカバレッジが低いため、特定地点での詳細な時間を指定することができない。例えば、JAXA OPEN API の利用データの一つである GCOM-W1 の場合、回帰日数が16日、概回帰日数 (地球上の同一地点の付近を再帰的に観測できる日数) が2日である。そこで、地球上の全観測地点を網羅するために、ユーザ指定時刻の前後1日を含んだ計3日間のデータを用いて指定領域の物理量の平均値を計算し、ユーザに対してはその平均値を提供することとしている。すなわち、ユーザが指定できるデータの空間解像度は、緯度・経度ともに0.1度 (赤道上で約11kmに相当) であり、時間解像度は1日 (前後1日を含んだ計3日間の移動平均) である。指定された地点の物理量が無い場合やトークンや日付のフォーマットが不正の場合は、error (対象データがない) を返す仕様となっている。

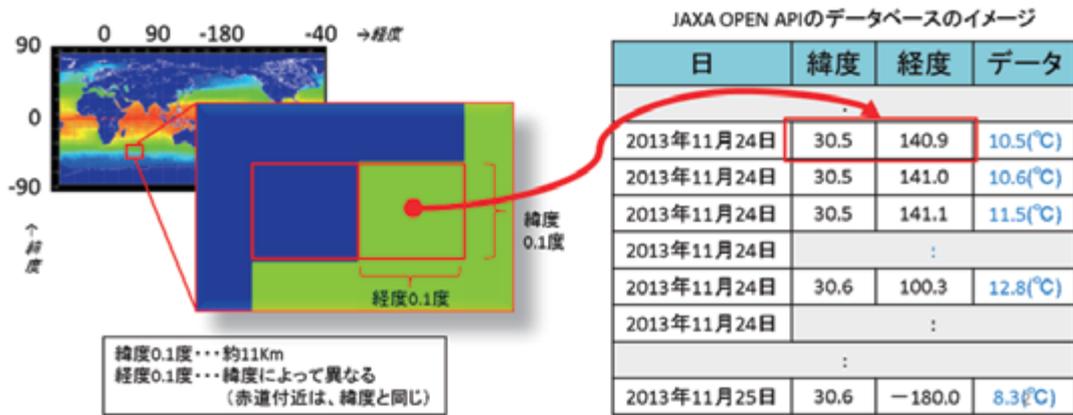


図 4 データ解像度のイメージ

2.4. データの指定

JAXA OPEN APIでは、リクエスト URL 及びリクエストパラメータに基づいて、HTTPS プロトコルの GET メソッドでデータを取得する。本システムが提供する地球観測データは、基本的には全球(海洋域・陸域)にわたって存在することから、ある一地点(緯度・経度)の情報を指定して取得するよりも、ある範囲に含まれるデータを一度にリクエストする方が、リクエストの回数を削減することができる。また、2.3 節でも述べた通り、ユーザが指定できるデータの空間解像度は、緯度・経度ともに 0.1 度(赤道上で約 11km に相当)である。そのため、任意の一地点の情報を取得しようとする、空間的に補間したデータを提供することになることから、データの精度を劣化させる原因となる。

そこで、本システムでは、ユーザがリクエストした緯度経度の範囲(0.1 度以上)に含まれる全データを取得する方式(all 指定)と、検索された全データの平均値を取得する方式(ave 指定)の二種類の取得方式を採用している。またこれらのデータ取得方式はリクエスト URL 上で指定することができる。例えば、リクエスト URL 上で、“<物理量指定>+all”を指定した場合は、指定した範囲に含まれる複数地点の物理量を全て返し、“<物理量指定>+avg”の場合には、指定した範囲に含まれる複数地点の物理量の平均値を返す。これら一連のデータ取得方法の詳細については、参考文献(2)を参照されたい。

3. JAXA OPEN API COMPETITION (データ利用コンテスト)

3.1. コンテストの概要

JAXA OPEN API COMPETITION とは、参加者が JAXA の地球観測データを使ってスマートフォンや WEB サービスなどのアプリケーションを開発し、その成果を競うというアプリケーション開発コンテストである。本企画では、JAXA が所有する膨大な地球観測データを、これまでそのようなデータに触れたことのない他分野の方が利用し、データの新たな利活用分野を創出することを目的としており、本目的を達成するために下記のアイデアを導入した。

- (1) ユーザが使いやすい形式で JAXA の地球観測データを提供する
- (2) 地球観測データの物理的な意味や解釈を正しく伝える場を提供する
- (3) 地球観測データの利用アイデアが生まれるきっかけとなる場を提供する

(1)については、2章で説明した JAXA OPEN API を利用することにより、ユーザが利用したいデータ（物理量）を使用したい範囲や時期を指定することにより、簡単に取得できる仕組みを導入した。また、(2)については、ホームページ²上で JAXA OPEN API の利用方法やデータの概要の説明資料を用意するとともに、「データ利用説明会（第一回：東京，第二回：筑波）」を開催し、コンテスト参加者と JAXA 職員が Face to Face で質疑応答ができる機会を提供した。さらに、地球観測データの利用アイデアが生まれるきっかけとなる場として、コンテスト参加者同士がチームを組み、チーム毎にデータ利用アイデアを議論して発表する「アイデアソン」を開催した。これらの一連のイベントを通じて、「API 利用登録」、「開発チームの編成」、「アイデアの具体化」、「アプリケーションの開発」がスムーズに実施できるような仕組みづくりを行った。JAXA OPEN API COMPETITION のイベント概要を図 5 に示す。

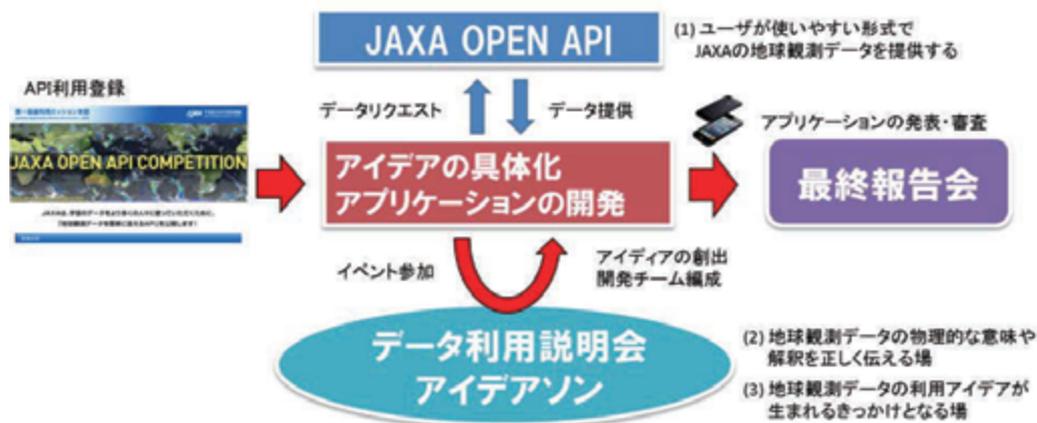


図 5 JAXA OPEN API COMPETITION の概要

なお、今回のコンテストでは、アプリケーション開発の目的（テーマ）を「安心・安全」と設定し、地球規模の課題解決をテーマとした地球観測データの利用アイデアの創出を目指すこととした。この安心・安全という言葉には、地球環境、自然災害、人間社会、情報ネットワークをはじめとする幅広い意味の「安心・安全」を含んでおり、地球観測データと我々の実生活とを結び付け、地球規模の課題解決を図るようなアプリケーションを募集した。さらに、これらのアイデアの創出にあたっては、「(1) 地球観測データの視覚化、表現方法の工夫」、「(2) 他の観測データや統計情報と組み合わせることによる新たな見方・考え方」等を盛り込みながら、これまでデータに触れたことのない方々の関心を引き付け、新たな気付きを生み出すようなアプリケーションの開発を目指した。

3.2. データ利用説明会

データ利用説明会は、コンテストの趣旨や API で提供する地球観測データの概要、さらには API の利用方法を Face to Face で解説することにより、コンテスト参加者の疑問や悩みを解消することを目的として開催した。2013 年 11 月 24 日（日本科学未来館）と 12 月 21 日（JAXA 筑波宇宙センター）の 2 回に分けて開催し、のべ 64 名の参加者と JAXA 職員との間で活発な意見交換が交わされた。

3.3. アイデアソン

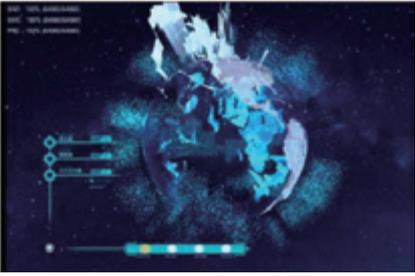
アイデアソンとは、コンテスト参加者同士の交流や参加者のアイデア出しをサポートするために、全員参加型のブレインストーミングや、チームを編成してアイデアの具体化を行うことを目的としたイベントである。このイベントでは、単にアイデアを具体化するにとどまらず、創出されたアイデアに賛同するメンバー同士で新たな開発チームを構成することも目的としている。アイデアソンは、2014 年 1 月 19 日にデジタルハリウッド大学（駿河台キャンパス）で開催し、30 名程度の参加者があった。

このイベントでは、簡単な自己紹介から始まり、各自でのアイデア検討、参加者同士のアイデア交換、アイデアの再考、気に入ったアイデアへの投票、チーム編成、チームでのアイデアの具体化を経て、各チームで検討したアイデアの発表と質疑応答を行った。このイベントを通じて、最終的に 5 つの地球観測データの利用アイデアが具体化され、このイベントで生まれたアイデアをもとに、コンテストに出品された作品も存在した。

3.4. 最終報告会

JAXA OPEN API COMPETITION の最終報告会は、2014 年 3 月 2 日に日本科学未来館にて開催された。最終報告会では、全 18 作品が各開発チームから発表され、その中から 3 名の審査員の推薦により、最優秀賞を含む 4 つの作品が優秀賞に選ばれた。優秀賞に選ばれた作品とその概要を表 1 に示す。

表 1 JAXA OPEN API COMPETITION 受賞作品

	<p>【最優秀賞】</p> <p>作品名：ぐるぐるアース ～水の惑星 地球～ 開発者：ぐるぐるアース開発チーム</p> <p>「水循環を直感的に理解」することをテーマとして、水に関わる地球観測データをグラフィカルに表現したアプリ。見ていて心地よい、美しい水の惑星地球を表現し、見ているだけでも癒されるアプリである。</p>
	<p>【JAXA 賞】</p> <p>作品名：グローバル都市マッチングシステム 開発者：グローバル都市マッチングシステム開発チーム</p> <p>世界の各主要都市の年間の降水量・積雪深・土壌水分量の変化から、類似のパターンを示す都市をマッチングし、気候の近い都市の組み合わせを探すアプリ。作付けするのに最適な農作物を調べる、防災対策や暮らしの知恵等を共有する等、産業や国際連携に役立てることを目的としている。</p>
	<p>【デジタルハリウッド大学賞】</p> <p>作品名：Personal Cosmos 開発者：Personal Cosmos プロジェクトチーム</p> <p>地球観測データを、デスクトップサイズの球状ディスプレイに、プロジェクターで投影（魚眼レンズに通して拡散）したアプリ。Leap Motion というセンサを利用して、手の動きによって映像を操作することも可能である。</p>
	<p>【日本気象協会賞】</p> <p>作品名：SMALL ACTIONS 開発者：PROJECT J</p> <p>世界中のユーザが、地球観測データを用いて、次の危険地域を予測するゲーム。予想を当てるとユーザのスマホには世界貢献の花が表示される。個別ユーザの予想座標は、現実のマップ上に表示され、実際の災害予想になる。予想情報は、SNS を介して、その地域のユーザに通知される。</p>

4. 結論

今回、我々は JAXA が所有する膨大な地球観測データを、これまでそのようなデータに触れたことのない他分野のユーザが利用できる仕組みを提供するために、地球観測データを「シンプルで直感的な形で」ユーザに提供することを目的とした WEB API (JAXA OPEN API) を開発し、2013 年 11 月より試行運用を開始した。また、本システムで提供されるデータを使ったスマートフォンや WEB サービスなどのアプリケーションの開発コンテストを開催した。コンテストでは、最終的に全 18 作品の地球観測データを利用したアプリケーションが提案されたことで、本 API とコンテストを通じて地球観測データとユーザが繋がり、それによって地球観測データの利用可能性が広がることが示された。

本サービスはまだ試行運用段階であり提供するデータ種類も限られているが、本サービスが提供する地球観測データは、他のデータにはないグローバルかつユニークな情報を含んでいる。そのため、本データと他のオープンデータとを組み合わせることで、これまでになく視野と視点で物事を見ることが可能になる。本サービスによって宇宙データとユーザが繋がり、さらにアプリケーション上で宇宙データと他のオープンデータが繋がることで、新たな価値が創出されることを期待している。

謝辞

本イベントは、JAXA 職員有志で結成された「新しい宇宙利用を拓く」ミッション検討チーム (Aseeds) によって実施されました。本イベントに協力して下さったデジタルハリウッド大学、日本科学未来館、日本気象協会の関係者の皆様に心より感謝いたします。

参考文献

- 1) JAXA OPEN API 利用データの解説, http://www.satnavi.jaxa.jp/jaxa_open_api/file/data.pdf
- 2) JAXA OPEN API の使い方, http://www.satnavi.jaxa.jp/jaxa_open_api/file/how_to_use.pdf

観賞性に重点を置いた「すばる」望遠鏡 HSC 画像の画像処理

上坂浩光*¹The image processing M31 shot by 'SUBARU' telescope with 'HSC'
which put emphasis on enjoyment

Hiromitsu Kohsaka

ABSTRACT

Astronomical images taken for scientific data analysis may not be necessarily “beautiful” for ordinary people. To appeal the beauty of astronomical images to ordinary people, a special kind of image processing is required, putting more weights on their visual appearance. I assume that such a kind of image processing is very different from that for scientific purposes. In this paper, I will present what kinds of image processing technique should be applied to scientific images to make them more appealing for public relations. As an example, I will use the Andromeda galaxy image taken by the Hyper-Suprime-Cam (HSC) equipped on the SUBARU telescope of the National Astronomical Observatory of Japan.

概要

科学的に正確なデータ解析を行うために取得された天体観測画像は、一般の人にとっては、必ずしも美しいと言えるものではない。広く一般の人にアピールするためには、美しさを持った「観賞用の画像処理」が必要である。科学研究を目的とする画像処理と鑑賞性を高めるための画像処理は異なると考える。本論文は、科学観測のために取得された画像を、広く広報に役立てるためには、どのような画像処理を行えば良いかを示すものである。その実例として、今回は国立天文台すばる望遠鏡、HSCによって取得されたアンドロメダ銀河の画像処理を例にとりながら、説明していく。

Keywords: Astronomical Image processing, Public Relations, Subaru Telescope, Visual Appearance

* 平成 27 年 12 月 17 日受付 (Received December 17, 2015)

1 はじめに

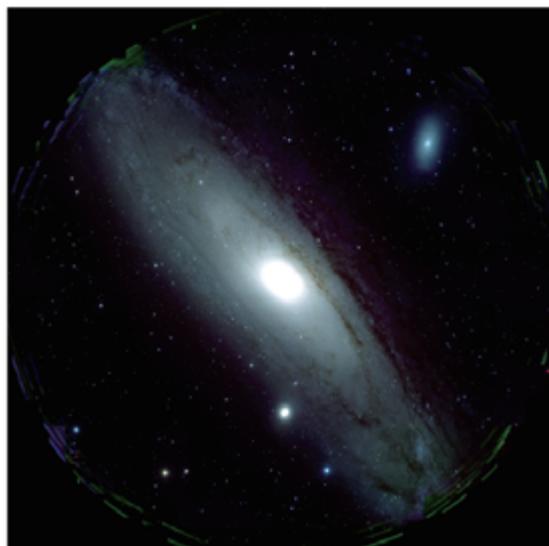
私は映像制作を仕事にしている関係上、非常に多くの「絵作り」を行ってきた。「人が何を美しいと感じるのか?」という問題と、常に向き合ってきたと言える。また自分自身でも天体を撮影し、写真展なども開いている。それらの経験を踏まえ、テクニックと感性の両面からこの問題に踏み込めればと考える。感性を数値化することは難しいところだが、なるべく客観的な評価を試みたいと思う。

2013年7月31日、国立天文台から、すばる望遠鏡に新しく設置された広視野カメラ Hyper Suprime-Cam^{*2} (以下、HSC)で撮影されたアンドロメダ銀河 M31 の画像が公開された (画像 1)。この画像は約月3個分 (視野角 1.5 度) の広がりをもつ M31 を、1 ショットで隅々まで鮮明に捉えた画像である (<http://www.nao.ac.jp/news/topics/2013/20130731-subaru-hsc.html>)。

しかし、確かに視野は広いものの、天体写真として見た場合、美しく感じなかった。銀河中心から周辺部にかけての階調が滑らかでなく、銀河のへりの部分には本来は無いはずの赤紫色の帯が現れていた。カラーバランスも不自然で、全体的にブルーかぶりを起こしているように見えた。銀河が発する色は全体的な総和としては白色光であり、特定の色に偏るものではない。コントラストも高めで、暗部が潰れてしまっている。私も含め、周囲から聞こえてきた感想は、「アマチュアが小さな望遠鏡で撮った画像みたいだ...」というものだった。参考までにアマチュアが 10cm の望遠鏡で撮影した画像を掲載するが、一見同程度の画像に見える (画像 2)。

広い視野を一度に撮影することができる新型カメラである HSC。その素晴らしさをアピールするために公開されたこの画像は、その本来の目的を果たす以前に、「美しくない」という目で見られてしまったことになる。広い視野、そして周辺部まで歪みのない M31 の画像。それは科学的にみれば十分な成果を体現していたが、一般の人にインパクトを与えられたかといえ、疑問の残るものだった。

それを残念に思った私は、すばる望遠鏡の広報担当者に連絡を取り、画像処理をやらせてもらえないかと申し出た。国立天文台内部でも、広報用画像としては不十分なものであるという認識があったようで、この申し出は好意的に受け取られ、すぐに画像の再処理をすることになった。



画像 1 : 国立天文台から発表されたすばる望遠鏡 HSC による M31



画像 2 : アマチュアが撮影した 10cm 反射望遠鏡による M31

2 観賞用画像処理の実際

2.1 画像サイズ調整

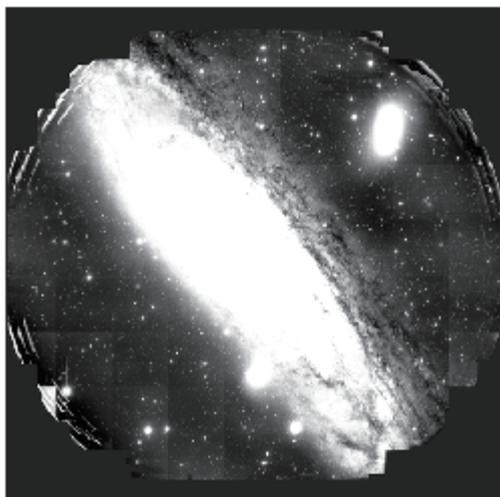
国立天文台から提供を受けた画像は、下処理（ダーク減算，フラット補正，コンポジット処理等を終えたもの）を終えた3枚のFITSファイルである。これらはgriバンドフィルターで撮影され、それぞれ5.1GBもある巨大なファイルだった。

まず困ったのはこのファイルを開く事が出来なかった事である。ステライメージ^{*3}、MaxImDL^{*4}などのアプリケーションは全く使えなかった。たまたま仕事で使っている会社のMacPro（主メモリー64GB）にインストールしてあったPixInsight^{*5}で試してみると、開くことが出来た。

その画像（画像2）は正に圧巻で、非常に高い解像度を持っていた。しかし、まだ試験観測中の為か、画素欠陥やフラット補正の甘さなどが目立っていた。画像にはCCDチップの四角いエリアが浮き出てしまっている。HSCのCCDカメラは、多数のCCDチップ（116素子）で構成されている。チップ毎の微妙な特性の差によってその境界が浮き出てしまうのであるが、本来であればフラット補正などによって抑えられるはずのものである。これを修正するには、補正処理をやり直す必要があるが、今回はこのデータ以前のものは提供されなかったので、あきらめざるを得なかった。つまりこの欠陥を隠すためには暗部を暗く落とす必要がある。国立天文台が発表した画像が、高いコントラストで暗部を潰している意味がよくわかった。

オリジナル画像は、35840x35588ピクセルのサイズがあり、このままだと、処理ができないので、25%の縮小を行い、8640x8640ピクセルにして処理することにした（画像3）。整数倍（25%）としたのは、画素補間による影響が出ないようにするためである。

また、縮小を行うと画素の平均化が行われノイズが減少する。低ノイズは美しい画像を得るための重要な要素であり、十分な解像度がある場合は、縮小処理によるノイズの低減は有用な方法である。



画像2：国立天文台から提供された下処理の終わった画像



画像3：25%縮小された画像の等倍画像

2.2 カラーバランスの考え方と問題点

今回の画像は、gri（緑，赤，赤外）フィルターにより撮影されたもので、そのままでは可視光色を再現することはできない。そこで g（緑），r（赤），i（赤外）を，b（青），g（緑），r（赤）に割り当て、擬似カラーで表現することにした。天体写真の世界では、水素輝線で輝く星雲は赤く表現されるのが常識だが、今回の処理画像では、r（赤）がg（緑）にマッピングされることから、輝線星雲が、緑に見えることになる。その点については今後の b（青）バンドでの追加撮影を期待したい。

さて、正しいカラーバランスは、対象に既知の色がある場合には容易に処理することができる。映画やビデオなどの撮影では通常白い紙を撮り、それが白くなるように補正をかける。しかし天体写真の場合には対象に完全な白を用意することはできないので、事前に補正値を求めておくことが重要となる。以下に補正する要素を述べるが、天体写真のカラーバランスは機器が持つ固定された特性だけでなく、天候など、環境によって変化するファクターがあることを忘れてはならない。

2.2.1 感度特性

CCD_rev：感度特性補正值

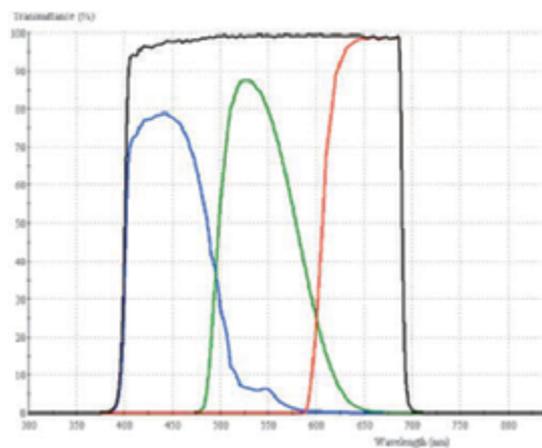
CCD チップの感度は、CCD チップ自体の感度と、それに組み合わされるフィルターの周波数特性によって決まる。

フィルターの周波数特性は、短波長の方が透過率が悪くなるのが普通である（bバンドは感度が低い）。これらの特性はフィルターの製造メーカーが提供してくれるので、そのグラフからおおよその値を読み取ることができる。余談であるが綺麗な色を表現するためには、これらフィルターの重なる部分が重要である。もし重なりがないフィルターを使うと、中間色が表現できずに色相変化の乏しい画像になってしまう。

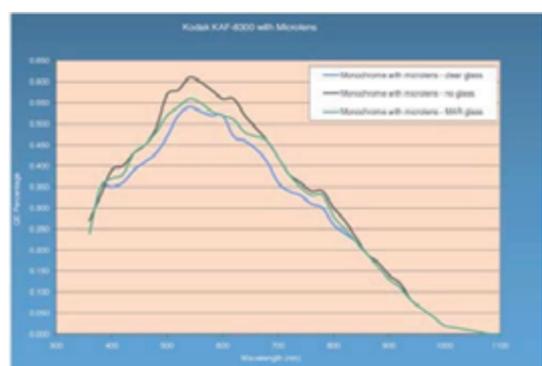
CCD 素子も、周波数によって感度が変化する。この特性もチップベンダーが情報を提供しているので、それを使用すると良い。

ここでフィルター毎の CCD の感度特性補正値を導き出すには、使用するフィルターの透過周波数範囲をサンプリングし、そのサンプリング周波数における CCD チップの感度特性とフィルターの透過率を掛け算することによって得られる。それらのサンプリング値を積算したものが、感度特性となる。

この方法はかなり煩雑であり、フィルターや CCD チップの製造上のバラツキの問題には対処できない。完全な白色光を用意しそれを撮影した画像を使って補正値を導き出すのが、もっとも正確である。



画像 4：「SBIG 社 LRGB」フィルターの周波数特性例



画像 5：「KODAK 社 KAF8300」 CCD 素子の周波数特性例

2.2.2 露出時間

EXP_rev：露出時間補正值

各フィルターでの露出時間が違う場合、それを正規化する必要がある。全てのフィルターで撮影時間を同じにすればこの補正項目は必要がないのだが、先に述べたように、通常、b（青）バンドのフィルターは透過率が低く、g（緑）やr（赤）と同じ露出を行うと、対象がそもそも写らないということが起こる。わずかでも写っていれば、それを画像処理で持ち上げることも可能だが、そのような処理はノイズを増加させるので、観賞用の画像を得ることはできない。よって観賞用の撮影の場合、フィルター毎の感度特性を考慮した露出時間で撮影する必要がある。論理的には、感度特性の逆数を乗算した値を使えば良いことになる。そのように撮影すれば「感度特性」と「露出時間」の補正は後処理で行う必要がなくなる。しかし、そのような計算で得られた露出時間は、秒単位以下の制御が必要になり、現実的ではない。通常の撮影では、5分、10分というようにキリの良い数字で行うことが多いので、そのような意味で露出時間の補正が必要になる。

2.2.3 撮影時の環境（空の透明度、光害の程度など）

ENV_rev：環境補正值

空のバックグラウンドの明るさや、撮影時に雲が通過した場合などはそれらが積算され、取得できる画像の輝度レベルは変化する。光害などでバックグラウンドが明るくなると、星の明るさとの差が小さくなる。通常バックグラウンドは黒に落とすので、結果的に星像が暗くなることになる。また、雲が通過した場合は、星の光を遮るので、星の輝度が落ち、さらにバックグラウンドも持ち上げられることからより大きな悪影響を与える。一見、雲のない快晴の空であったとしても、大気中の水蒸気の量によって、これらの現象は少なからず起きていると考えられるので、まったく同じ条件での撮影はできない。この補正項目は、固定できず、また数値化して補正するのも困難である。天体写真撮影にとって重要なファクターである。

*

以上の事から、正しいカラーバランスを得るための補正值は、各バンド毎に、以下のようなものになり、この補正值を、取得画像のピクセル値に対して乗算を行ったものが、補正画像となる。

＜各バンドに対するカラーバランス補正值＞

$$\text{Rev} = \text{CCD_rev} \times \text{EXP_rev} \times \text{ENV_rev}$$

$$P' = P \times \text{Rev}$$

<p>Rev：カラーバランス補正值 CCD_rev：感度特性補正值 EXP_rev：露出時間補正值 ENV_rev：環境補正值 P：元画像のピクセル値 P'：補正されたピクセル値</p>

この中で、「感度特性補正值」は固定値であり、正しい方法論に則って導きだせば問題ない。「露出時間補正值」は、撮影時に確定する値だからこれも問題ない。しかし「環境補正值」は、その時の撮影条件によって変動しあらかじめ決めておくことはできない。よって、この問題を解決するには、撮影された画像から「環境補正值」を導き出すなんらかのアルゴリズムが必要になる。このような補正の考え方は観賞用天体写真のみならず、科学的観測にも有用かもしれない。

2.3 「AutoStretch」

そこで私が発案したのが、「AutoStretch」と呼ばれるアルゴリズムで、取得した画像から前章であげた「EXP_rev：露出時間補正值」と「ENV_rev：環境補正值」を導きだす。この2つの処理を行うことを画像の正規化と呼ぶことにする。こうして正規化した画像に「CCD_rev：感度特性補正值」を入力してやれば、正しい発色が得られることになる。

このアルゴリズムの基本的な考え方は、「どんな条件で撮影された画像も画素値の標準偏差（ばらつき）は同じになるのではないか？」というものである。加えてバックグランド値を合わせる（背景を黒にする）機能も持たせることにした。以下のような式でテストしてみると、良好な効果を得ることができた。

< AutoStretch 計算式 >

$$P' = (P - BgArea) \times (StdDevR \times Rev) + Offset$$

$$StdDevR = TStdDev / StdDev$$

P:元画像のピクセル値

TStdDev:基準画像の標準偏差

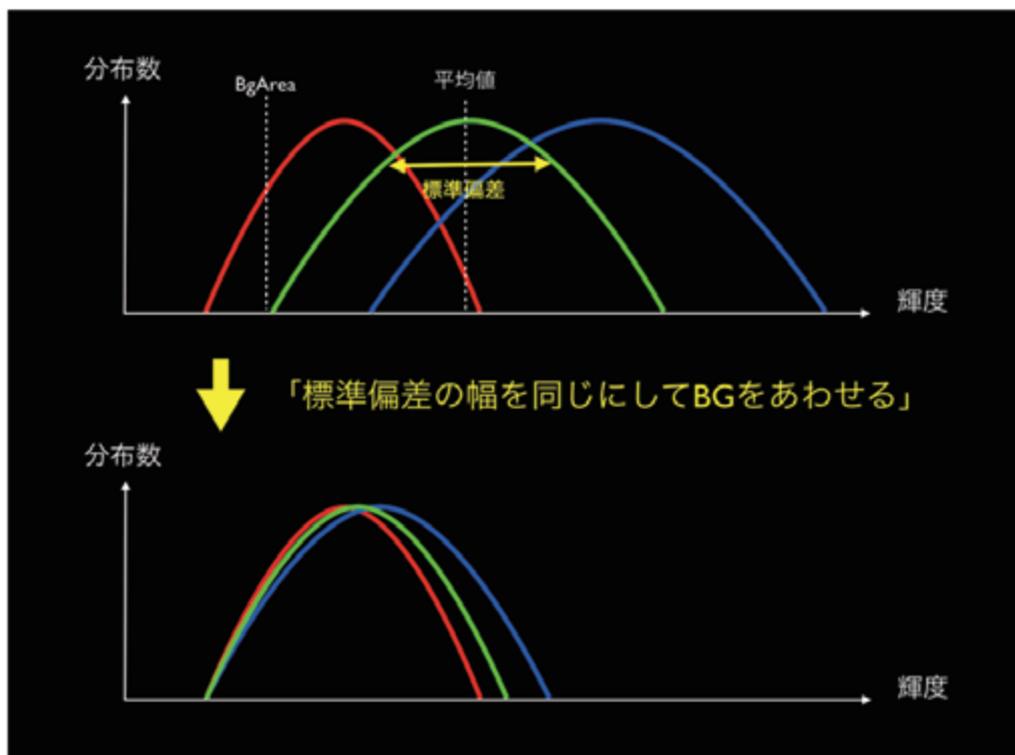
StdDev :元画像の標準偏差

BgArea :指定された背景（バックグランドエリア）の平均値

Offset :ユーザーが与えたオフセット値

Rev:補正值

ここでやっていることは、「バックグランド値を合わせ、標準偏差値により正規化を行い、最後にフィルターの露出倍数を乗算する」（画像6）ことである。これによって背景が完全な黒になり、光害値などで撮影された天体写真も色かぶりなく綺麗に表現することができた。



画像6：AutoStretch の処理概念図

以下が実際にこのアルゴリズムで処理した画像例（画像 7）である。補正なしでは、緑の色かぶりが見られ、背景の黒に緑がのっているが、正規化するだけでも、大きな効果を示していることがわかる。そこに感度特性補正値を加えることによって、美しい色が再現される。



画像 7 : AutoStretch による処理画像の比較

このアルゴリズムは、アストロアーツ社が販売している「ステライメージ」にも搭載され、バックグラウンドエリアの自動サーチ機能や、フィルター補正値の効果をリアルタイムに確認しながら調整を行うことができるなど、より発展された形で実装されている。ただし、全画面を覆い尽くす真っ赤な星雲など、色相が極端に偏る対象の場合にはうまくいかない場合がある。この場合は感度特性補正値の値を手動で操作し、調整する必要が出てくる。

2.4 輝度とカラー情報を分けて画像調整する

人の眼は、輝度に比べて色に対する分解能が低い^{*11}と言われている。この原理を使い、動画像の記録には、輝度と色に分けた記録方式が使われ、この色の情報量を減らすことで、画質を落とさずに記憶容量を減らすなどの工夫がされている^{*12}。これと同様の考え方を天体写真画像処理にも取り入れて処理を行った。輝度情報と色情報を分けてそれぞれに最適な処理を施すことによって様々なメリットが生まれる。

まずはじめに、AutoStretch によってカラーバランスをとった gri 画像から輝度画像を作り、輝度と色、それぞれの画像に対して以下のような指針で処理を行う。

輝度画像：「シャープな画像処理を心がける」「暗部を潰さずに明るく表現する」

色画像：「画像をぼかして SN を上げる」「暗部を暗く落として黒にしてしまう」

このような処理をした後、Photoshop 上で、色画像を Lab（色差）モードに変換し、この L（輝度）チャンネルを輝度画像で置き換える。こうすることによって、高い解像度をもち、かつカラーノイズの少ない画像を得ることができる。

以下、輝度画像と色画像について、細かく説明する。

2.4.1 輝度画像

輝度画像は、gri バンド画像、3 枚をコンポジットし作成する。コンポジットすることにより、SN が向上し、滑らかな画像を得ることができる。SN は、コンポジット枚数のルートに比例するので、この処理で約 1.73 倍の SN を得ることができる。

<コンポジットによる SN アップの効果>

$$SN' = \sqrt{N} \times SN$$

SN' : コンポジット結果

N : コンポジット枚数

SN : 一枚の画像の SN

こうして得られた画像は、白飛びした画像になる。これは 16 ビットで撮影された画像を 8 ビットのダイナミックレンジしか持たないモニターで見ているためである。今後は HDR ディスプレイなど、元の画像が持つダイナミックレンジをそのまま表現出来るデバイスが増えてくるかもしれないが、現時点では、16 ビットの階調を調整し、8 ビットの中に収める（マッピングする）必要がある。

これを行うには、画像処理ソフトのトーンカーブ調整で高輝度部のカーブを寝かせる処理を行うのだが、非常に微妙な操作が要求されるため、デジタル現像*⁶ という数学的にトーンカーブを調整する処理を行った。この画像処理は、アマチュア天体写真家の岡野邦彦氏によって考案されたもので、フィルムの現像処理で起きている現象をデジタル的に行うものである。単にトーンカーブを調整するのではなく、高輝度部に対してのみシャープネス処理を同時に加える。

私の今回の画像では、後処理によって調整を行うために、シャープネス処理は OFF にし、トーン調整機能のみを使った。（画像 8）



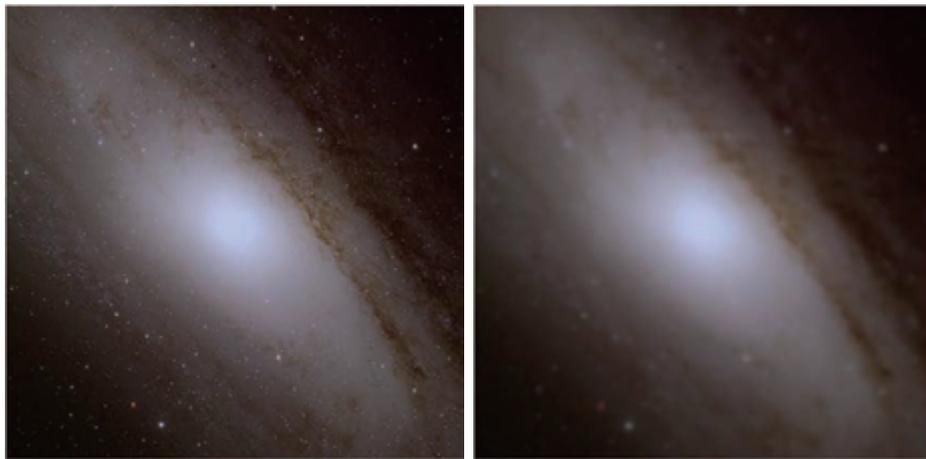
画像 8 : デジタル現像による階調処理の結果

こうして階調を整えた画像にシャープネス処理を行う。マルチバンドシャープ*⁷ や、画像復元処理*⁸ を行い、解像度感を高めるのだが、これらの処理は基本的に SN を劣化させる。しかし gri バンド画像 3 枚のコンポジットを行っているので、その SN 向上分を解像度アップのダメージに充てることができる。

今回の HSC による M31 の場合は、既に十分な解像度であったために、微小なマルチバンドシャープ処理を施した。

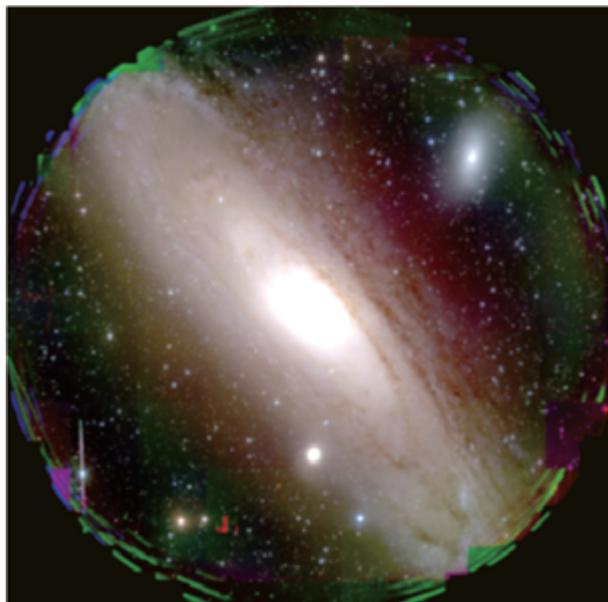
2.4.2 色画像

AutoStretch で得られた gri 画像を RGB チャンネルとして色画像化する。この色画像はぼかし処理を行い、SN を高める。ぼかすことにより失われたディテールは、後で行う処理 (Lab 合成) によって元どおり (輝度画像に施した処理によってさらに高解像度) になるので、完全にノイズが消えるまで思い切ってぼかしてしまっても良い。(画像 9)



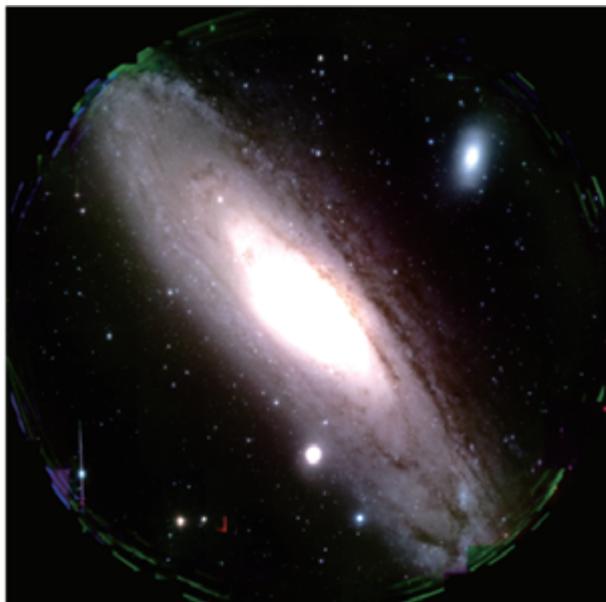
画像 9 : 左) オリジナルの色画像. 右) ガウスぼかしフィルタ

私は画像処理の途中で、輝度と彩度を思い切り持ち上げてみるということをよく行う。こうすることによってその画像の欠点を確認することができるからだ。この色画像に対してもその処理を行ってみるとその欠点があらわになった。様々な要因によってフラット補正ミス*⁹ が起こっており、M31 の背景に CCD チップの境界が浮き出てしまっていたり、M31 のへりにも赤紫の色がついたりしている。国立天文台が発表した画像の「銀河のへりの赤紫色の帯」はこれが原因だったと考えられる (画像 10)。



画像 10 : 輝度と彩度をもち上げたチェック画像

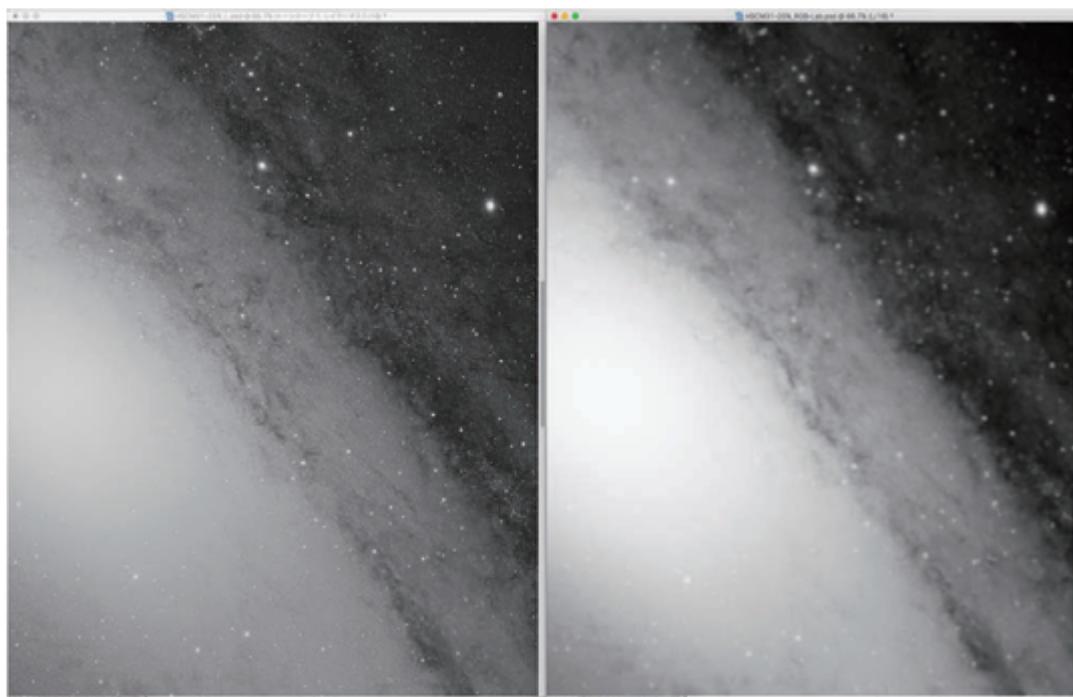
そこで、背景を完全な黒になるよう、暗部を落とすことにした。トーンカーブ等の調整である輝度以下の部分を黒くして潰してしまうのだ。こうすると M31 の淡いガスの広がりの部分にも色がのらなくなるのだが、先にも書いたように「人の眼は低輝度部分に色を感じない」という特性を持っているので、逆にその方が自然な画像になると思われる。（画像 11）



画像 11：補正後の色画像（チェックの為の輝度，彩度強調ありの状態）

2.4.3 Lab 合成

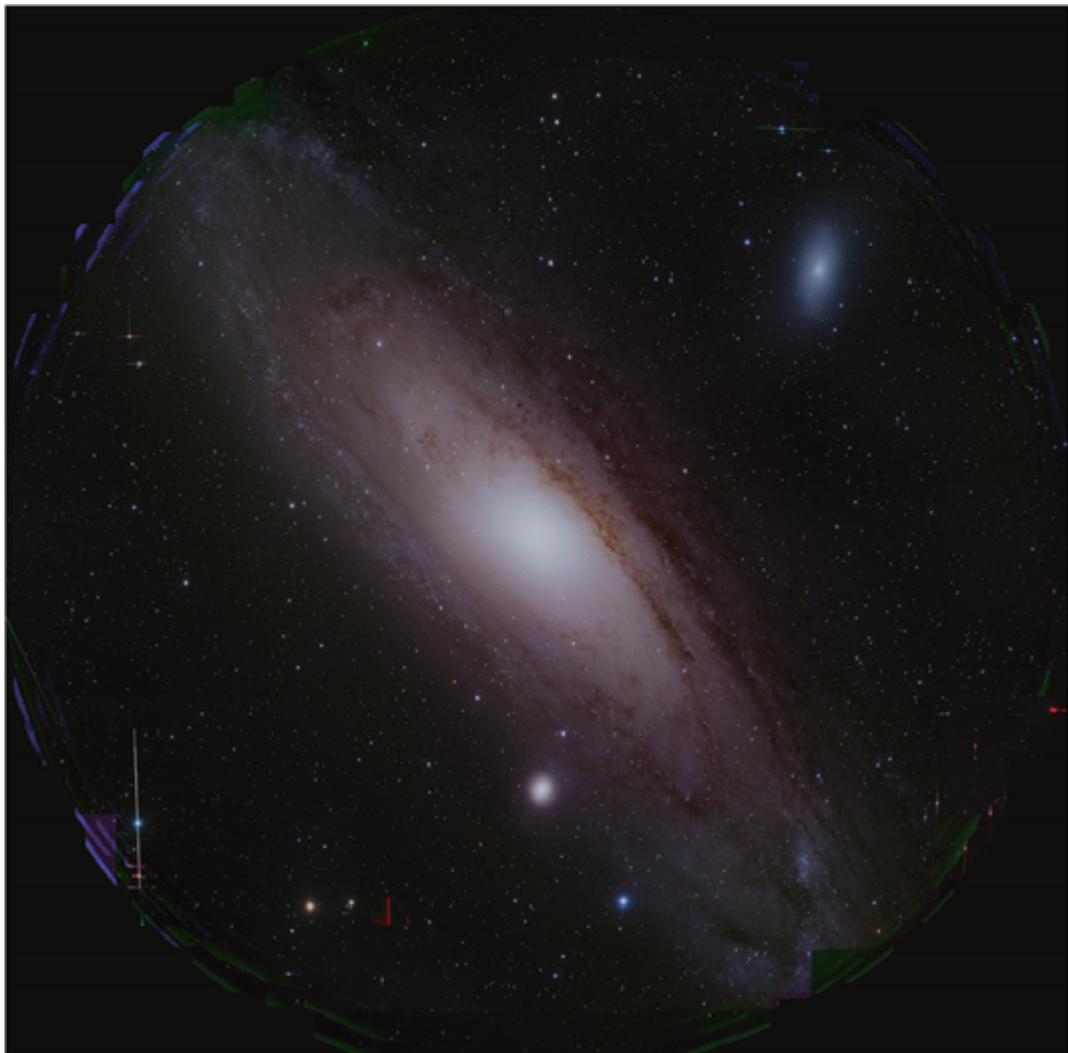
こうして調整を終えた色画像を Lab 色空間に変換する。Lab 空間^{*13}とは、補色空間の一種で、明度を意味する次元 L と補色次元の a および b から成り立っている。つまり輝度情報 (L) と色情報 (a, b) を分けて扱える色空間である。そしてこの色空間の輝度情報 (L) を、同じく調整後の輝度画像で置き換えることによって、一度ぼかしてしまった解像度をより良い状態に戻す効果を得る。これを Lab 合成と呼ぶ。（画像 12）



画像 12：左) 輝度画像で，右) Lab モードにした色画像のぼけている L チャンネルを置き換える。

この時注意しなければならないのは、置き換える輝度画像は、Lab モードにした色画像の L チャンネルとある程度同じ調子をもつように調整する必要があるということである。極端に調子が違うと良い結果は得られない。その為には、色画像の調整を先に行い、それを Lab モードに変換して、L チャンネルの画像を確認しておく方法が良い。この L チャンネル画像を見ながら、輝度画像のデジタル現像を行うことによって、同じ調子の輝度画像を作ることが出来る。以上の操作が終わったら、RGB モードに戻す。

(画像 13) がこのような処理をした画像である。元の画像にあった背景のむらや緑かぶり、M31 周辺の赤い部分が消えているのがわかる。また、この画像サイズではわかりにくいが、解像度感も元の画像よりアップしている。ここまでの処理で、さまざまな欠点が克服されているのは確認できたがこれで完成として良いだろうか？



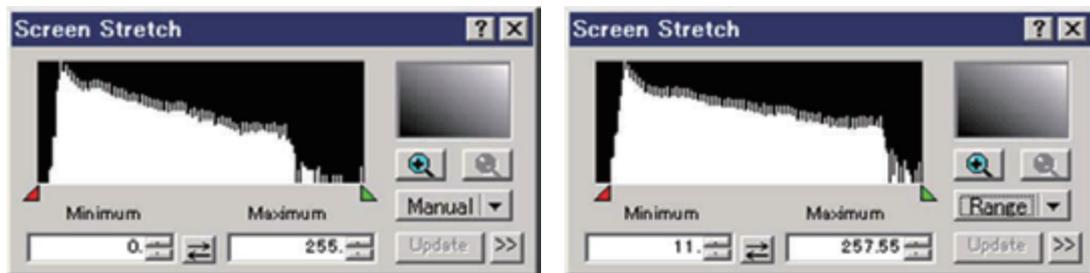
画像 13：L チャンネルを輝度画像で置き換え、RGB モードに変換した画像

2.5 画像調整「人は何に美しいと感じるのか？」

画像の観賞性を最大限高めるためには、今まで行ってきたような論理的な調整だけでなく、最終的に人の感性による調整が肝要である。このような調整は一般の写真でも行われているが、天体写真は特にそれが難しい。それはその対象を誰も見たことがないからである。写真は通常、実際に見たイメージに近づけるように調整していくが、天体は直接自分の眼で見ることができないために、何が正解かがわからない。試しに Web で M31 の画像を検索してみると、様々な色合いの沢山の M31 が現れる。つまり、調整する人の個性によってこれほど結果が違ってしまふのだ。これはセンスの差としかいいようがない。では、そのセンスはどのように形成されるものか？ 私はその人が、それまでに生きてきた中で見た様々な風景、受けた経験（感動）によって出来上がるのだと思っている。人は無意識に美しいもの、そうでないものを分類している。そうして蓄積した記憶が、その人の「美」という感覚になるのだと思う。これだけで人と人それぞれの「美」が完成してしまうが、幸いなことに、私たちは地球上の風景を共有している。つまり、それによって共通の「美」にたどり着くのではないかと思っている。

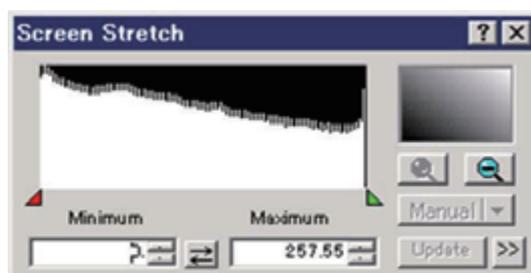
さて、そんな自分の「センス」で、今回の Lab 合成の終わった画像（画像 13）を見てみると、全体に調子が鈍く、輝度感がないと感じる。また全体的に緑かぶりしているようにも見える。細心の注意をもって進めてきた処理であったが、人はわずかなバランスの狂いを見抜く力をもっている。

そこで Photoshop を使い、これらの欠点を取り除いていった。このような「感性による処理」を数値的に表現することは困難であるが、輝度感の調整に関して言うと、調整前と調整後の画像ヒストグラムには、以下のような変化が見られた。



画像 14 : 左) 調整前 (画像 13) 右) 調整後 (画像 16) のヒストグラム *高さ方向は LOG

調整前のヒストグラムは、高輝度部分のピクセル数が少なく、全ダイナミックレンジを使い切っていないように見える。調整後は最大輝度（右側）ギリギリまでヒストグラムの壁が伸びている。このあたりが心地よい輝度感なのかもしれない。参考までに、国立天文台の画像のヒストグラムは以下のようなになる。最大輝度のピクセル数がこんなに多いということは白ベタの領域が大きく存在していることを示す。美しい画像の最大輝度ピクセル数は少ない方が綺麗に見えると言えるだろう。また、最低輝度（真っ黒）も多数存在している。私が処理した画像には、最低輝度周辺には隙間があり、真っ黒なピクセルが一つもないことも大きな違いと言えるだろう。



画像 15 : 国立天文台の画像 (画像 1)

輝度調整以外では、背景に残ったわずかな色を、M31の輝度情報から作成したマスクを使って、彩度を下げた。これはフラット補正ミスを隠すためではあるが、ローカルな処理（部分的に画像を変更すること）を行ったことになる。科学的な画像ではこうした処理は許されないが、観賞用の画像ではその規則を破って良いと考えている。

また、このような最後の微調整には十分な時間をかける必要がある。人の視覚には恒常性*10があり、あまりにも長く同じ画像を見続けると、良いかどうかを正しく判断できなくなることがあるからだ。私の場合、最後の確認は、1日程度時間をあけて行うようにしている。（画像16）



画像16：完成したM31画像

3 まとめ

完成した画像（画像 16）は、縮小されると良くわからないが、大きくプリントすると、驚く位のディテールを見ることが出来る。この完成画像は、国立天文台と私の著作物となり、観賞用の画像処理に対する価値観を認めていただいたと認識している。そして、すばる望遠鏡 HSC の広報用画像として、様々な場面で使われることとなった。2014 年 10 月 28（火）～2015 年 2 月 22（日）に国立科学博物館で開かれた「ひかり展」には、大きく引き伸ばされたこの画像が展示され、多くの人が目にするようになった。

今回アマチュア天体写真家が、すばる望遠鏡の画像処理を手伝うという貴重な機会を得られた事は画期的だが、本来は国立天文台内部に、観賞（広報）用の可視化を行う専門部署が設けられることを願っている。鑑賞用の天体画像は、ハッブル宇宙望遠鏡の画像を例にとるまでもなく、一般の人に宇宙の姿を訴える上で、とても有効な手段だ。これは国立天文台のみならず、様々な科学に取り組む研究機関全てに言える事だと思う。研究本来の目的を果たした画像が、美しい観賞用の画像に生まれ変わったら、多くの人にその研究をアピールすることができる。その価値を認識し、そこにもきちんと予算をかけることが、今後の科学の発展にとって大切な事だと考える。

- 1) 上坂浩光：有限会社 ライブ Hiromitsu Kohsaka:Live Company Ltd. 代表取締役・映画監督・CG クリエーター・天体写真家。天体写真に関しては「星居 Web」で作品を見ることが出来る。「星居 Web」：http://www.live-net.co.jp/SeikyoWeb/Seikyo_Web_Site/Welcome.html
- 2) Hyper Suprime-Cam（ハイパー・シュプリーム・カム）：2012 年 8 月 17 日にすばる望遠鏡に設置された広視野高解像度カメラ。高さが 3 メートル、重さが 3 トン、第一レンズ直径約 82 cm、レンズ筒の長さは 165 cm の高性能フィールドフラッターに、高感度高解像度 CCD（約 8 億 7000 万画素）を組み合わせた視野角 1.5 度を誇る広視野カメラ。
- 3) ステライメージ：アストロアーツ社が開発した天体画像処理ソフト。国産ソフトでありサポートが充実している。
- 4) MaxImDL：カナダの Diffraction Limited 社が開発した天体写真撮影&画像処理ソフト。アマチュアの間では世界で一番普及しているソフト。
- 5) PixInsight：スペインの Pleiades Astrophoto 社が開発した天体画像処理ソフト。Windows 版、Mac 版、Unix 版と様々なプラットフォームに対応している。オープンソースによって多くの開発者が機能を追加し続けている。
- 6) デジタル現像：「フィルムにおける画像工学のデジタル画像への適用～冷却 CCD による天体撮像での応用例～
https://www.jstage.jst.go.jp/article/itej1997/57/2/57_2_203/_pdf
- 7) マルチバンドシャープ：画像を周波数ごとに分解し、それぞれに対してシャープネス処理を行う処理。画像の詳細さ毎にシャープネスの強さを変えられるので、天体写真に最適である。
- 8) 画像復元処理：http://www.live-net.co.jp/SeikyoWeb/Seikyo_Web_Site/PixInsight_DEC_01.html
- 9) 背景は画像のフラット補正が完璧でないとき色がついてしまう難しい領域である。現実的に完璧なフラット補正は不可能なので、こうして背景の色を強制的に黒にしまうことで、それを防ぐことができる。
- 10) 視覚の恒常性：視覚で捉えられる実際の画像（形、色など）を、補正しようとする生理学的・認知科学的な作用。夕焼けの光の中でも植物の葉は緑に見えたり、映画館でスクリーンに近いところでみてもスクリーンを台形に感じなくなるなどの作用。

- 11)12) コンポーネント映像信号 - Wikipedia <https://ja.wikipedia.org/wiki/コンポーネント映像信号>
- 13) Lab 色空間：補色空間の一種で、明度を意味する次元 L と補色次元の a および b を持ち、CIE XYZ 色空間の座標を非線形に圧縮したものに基づいている。 <https://ja.wikipedia.org/wiki/Lab色空間>

