

SPICE を用いた視野シミュレータ FLOW の開発

山本 幸生^{*1} 岡田 尚基^{*1} 本庄 英司^{*2} 大友 翔一^{*3}

Development of SPICE-based Field Of View Visualizer: FLOW

Yukio Yamamoto^{*1}, Naoki Okada^{*1}, Eiji Honjoh^{*2}, and Shoichi Otomo^{*3}

Abstract

The function to simulate an instrument's field of view has been widely used in past missions, and will be used in future missions. We develop an FOV simulator, FLOW, based on SPICE Toolkit for versatile application. The FLOW is composed of two console programs: one is a simulation engine, and the other is an image generator. The simulation engine outputs an XML file which includes a simulated result. The image generator reads the XML output, and generates an image in FITS/PNG/JPEG formats. In this report, we describe the software requirement based on use-case of the FOV simulation, and FLOW's functions to draw. In addition, introduce a web-based FLOW application using FLOW.

Key Words: SPICE, Field Of View, FOV, Visualization

概要

惑星探査機に搭載された観測機器の視野をシミュレートすることは、これまで多くの観測機器が行ってきたことであり、また将来的にも利用されつづける機能の1つである。私たちは、SPICE Toolkit を利用した、汎用的な視野画像シミュレータ FLOW を開発した。FLOW は2つのコンソールプログラムから構成され、一つはシミュレーション・エンジンであり、もう一つは画像生成プログラムである。シミュレーション・エンジンは、シミュレーション結果を含んだ XML ファイルを出力する。画像生成プログラムは、その XML を読み込み、FITS/PNG/JPEG フォーマットで画像を生成する。本報告では、ユースケースに基づいた要求項目を列挙し、FLOW の持つ描画機能について説明する。また FLOW を用いたウェブブラウザベースのソフトウェアについても紹介する。

1 はじめに

惑星探査において、観測機器の視野をシミュレートするツールの需要は多い。多くの惑星探査機にはカメラが搭載されており、実際に撮像した画像とシミュレートした画像を比較すれば、位置や姿勢、カメラの歪みの検証を行うことが可能である。

またシミュレーションでは、実際には起こりえない条件を設定することができる。例えば惑星の夜側を撮像した場合、実際の画像では真っ暗な画像が得られることもある。このような場合に、シミュレーションでは影の効果をなくし、撮像領域を明確にすることもできる。他にも、実際には撮像していない時間帯を模擬したり、予測軌道や姿勢を与えることにより、将来取得されるであろう画像を事前に作成することもできる。

目的に応じて表面に貼り付けるテクスチャ画像を変えることで、通常的地表面の代わりに経緯度線や白地図を用いたり、空間分解能の低い機器がどの領域を撮像しているのかを示したりすることもできる。

^{*1} 宇宙科学研究所 学際科学研究系
(Department of Interdisciplinary Space Science, Institute of Space and Astronautical Science)

^{*2} 株式会社アストローツ (AstroArts Inc.)

^{*3} 慶應義塾大学 (Keio University)

上記のような視野を模擬したシミュレーションは、これまで幾度となく行われてきたはずである。それにも関わらず、汎用的なツールとしては開発されてこなかった。本開発では、多くの観測機器に対して、最小限の設定を行うことで、汎用的に利用可能な視野シミュレータを開発することを目的とする。

汎用的な視野シミュレータの開発において、どこまで再現性を高める必要があるのかは検討が必要である。繊細なシミュレーションを行おうとした場合には時間がかかり、また適用すべきシミュレーションモデルの検討も必要となってくる。そこで本開発では、主なユーザとして惑星探査機に搭載する視野角を持つ機器の開発者を想定し、探査機運用や地上データ処理で利用可能なツールの開発を行う。

2 SPICE

観測機器の視野に含まれる映像をシミュレートしようと考えた場合、ある時刻に対して、探査機の位置、姿勢、観測機器の視野を考えて計算を行う。この計算を行う上で最大の関門は、必要な情報をどのように取得すればよいか、という点である。特に汎用的なツールとして開発する場合、これらの情報が画一的なフォーマットとして用意されていなければ、ツール側で多様なフォーマットに対応しなければならない。

幸いにして、惑星探査においては SPICE という汎用的なツールがある [16]。SPICE は、NASA のジェット推進研究所 (Jet Propulsion Laboratory; JPL) によって開発され、探査機の時刻、軌道、姿勢や、観測機器の視野角などの補助情報をアーカイブするために、SPICE カーネルと呼ばれる画一的なフォーマットを規定している。またそれら进行操作するためのライブラリ (SPICE Toolkit) の提供を行っており、ライブラリを通して SPICE カーネルを参照し、それをを用いた計算を行うことができる。例えば、ある時刻における探査機の姿勢や視野角の情報を得たいと考えた場合、SPICE カーネルと簡単なプログラムを用意すれば良い。必要な SPICE カーネルの多くは、SPICE のウェブサイトからダウンロードすることができる。

日本の惑星探査においても、姿勢や軌道などの情報は SPICE カーネルとして提供している。SPICE は今後の惑星探査で標準的に利用されると考えられるため、視野を計算するのに必要な SPICE カーネルが揃っている探査機においては、本シミュレータで可視化できることを目標とする。

3 シミュレータ概要

3.1 背景

本シミュレータは、視野を可視化するためのソフトウェアとして、2009 年から開発が始まり、当時は mpsim という JAXA 内のプロジェクトで使用するためのツールであった。翌年 2010 年の開発により、現最新版とほぼ同等の機能を備えたツールへと進化した。2012 年 12 月に、FieLd Of View Visulaizer から FLOW と名付け、ソフトウェアの他に、コンパイル環境や使用方法を詳細に記したウェブサイトを整備した上で一般公開を行った。FLOW のソースコードはインターネットからダウンロードすることができる [10]。

3.2 構成

FLOW は、シミュレーション・エンジン (flow_se) とイメージ・ジェネレータ (flow_ig) の 2 つのコンソールプログラムから構成される (図 1)。

flow_se では、設定ファイルである config file に書かれた条件を読み込み、探査機に搭載された機器の視野内に写る天体の座標を計算し、天体の位置情報をシーンファイルとして出力する。シーンファイルはそれ単体で描画に必要な情報を全て有しており、flow_ig 以外のアプリケーションからも利用可能なよう、汎用性の高い XML フォーマットを採用している。またシミュレーション結果に対して意図的にデータを挿入したい場合には、シーンファイルを編集することで、レンダリング結果に反映することができる。

flow_ig では、シーンファイルを入力とし、レンダリングを行い画像を出力する。このとき「恒星のスペクトル型によるカラー描画」「惑星・衛星のテクスチャマッピング」「形状モデルの描画」「観測機器の視野形状によるマスク」処理を行っている。

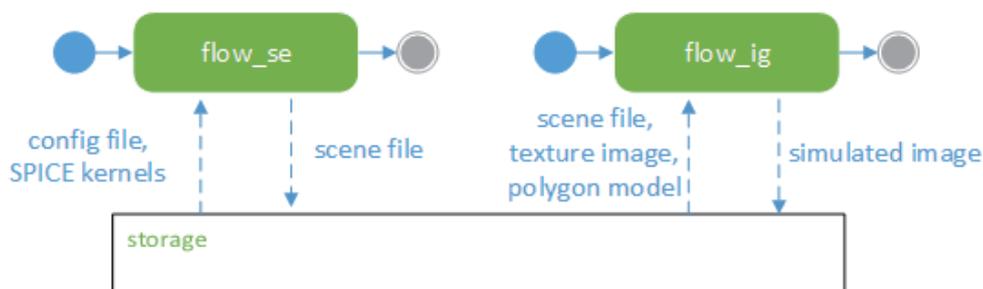


図1 視野シミュレータ FLOW の処理フロー。シミュレーション・エンジン flow_se が config file および SPICE kernel を読み込み scene file を作成する。イメージ・ジェネレーター flow_ig は作成された scene file と惑星等に貼り付ける texture 画像, polygon モデルを読み込んでシミュレーション画像を作成する。

FLOW の利用者が用意すべきファイルは、flow_se の入力となる config file と、計算に必要な SPICE カーネル群、描画に必要なテクスチャ画像、小惑星等の形状モデルである。

4 要求仕様

本ソフトウェアの開発を行う際の要求仕様を表1にまとめた。以下に各要求仕様について記述する。

4.1 マルチプラットフォーム対応

要求仕様の中で、特に優先度を高く設定したのは、マルチプラットフォーム対応である。SPICE Toolkit が正式に対応しているプラットフォーム上で動作することを目標としている。より多くのプラットフォーム上で動作させるために、特殊な環境が不要なコマンドラインベースの「コンソール・プログラム」を採用した。

現段階では POSIX[9] に準拠したアーキテクチャに対応している(表2)。POSIX 環境ではないアーキテクチャについては、未サポートとしている。また未サポートではないが、環境そのものが準備できなかったため、動作検証を行っていないアーキテクチャについては未検証とした。

4.2 設定ファイル

ユーザがシミュレーションを行う場合、全ての条件は設定ファイルである config file で行う。図2に実際の config file の例を示す。本 config file には、

- SPICE カーネルのファイルパス
- 星描画時のマグニチュード上限
- SPICE の計算条件 (時刻, 観測機器, 光路差の処理方法)
- テクスチャ情報
- 描画オブジェクト
- 画像のサイズ

が含まれている。

表1 要求仕様

No.	分類	要求詳細	理由
1	マルチプラットフォーム対応	SPICE Toolkit が動作するアーキテクチャで動作すること.	SPICE Toolkit は Mac, Linux, Windows(cygwin), Solaris, およびそれら環境の各種 32bit/64bit 環境で動作するため, 同環境で動作することが望ましい [18].
		コンソールプログラムとして動作すること.	X-Window や OpenGL などがインストールされていなくとも, シミュレートした画像を生成できるようにする.
2	描画情報の入出力	Lua 形式による設定ファイルを入力とすること.	複雑な設定をコマンドラインではなく設定ファイルから読み込むことで, 再現性を高めるようにする.
		XML 形式による描画情報を出力すること.	シミュレートした結果は, 各種情報をテキストファイルとして出力し, ユーザが確認できるようにする.
		天体情報を出力すること.	視野内に含まれる天体情報を出力する. また画像上での位置も含める. 例えば恒星の場合, Hipparcos カタログ [20] が存在し, その中には恒星までの距離やスペクトルなどが含まれている.
3	描画コンテンツ	ユーザが予め用意したテクスチャを, 対象天体の表面に貼付けられること.	月面などは詳細なテクスチャ画像により, 描画領域が正しいかどうか確認することができる. 部分的に高解像度のテクスチャを持つ場合, それにも対応する.
		リングを持つ天体を描画できること.	木星や土星を描画する際に, リングをテクスチャとして用意することで, リングのある惑星を描画する.
		恒星は, 色・明るさを考慮して描画すること.	SPICE カーネルには, Hipparcos カタログも含まれ, その中には色情報も含まれており, それを参照する.
		部分的なテクスチャを描画できること.	惑星全球のベースマップとは別に, 部分的に高解像度テクスチャが存在する場合, それを描画できること.
		形状モデルを表示できること.	小惑星など形状モデルが存在する場合, それを表示する. フォーマットは STL[23], SPICE DSK[17] に対応する.
		多角形・四角形・円・楕円の視野形状に対応すること.	例えば Venus Express の VMC は八角形の視野角を持ち, 視野領域以外をマスクする機能が必要である.
		プッシュ・ブルーム方式の撮像に対応すること.	視野定義が 4096x1 のようなラインセンサとなるため, そのまま描画すると細長い画像ができてしまう. そのため探査機の動きに合わせた画像を作る必要がある.
4	描画オプション	陰影の有無を制御できること.	例えば夜側を撮像した場合に, 影の効果を無視して描画を行う.
		ワイヤフレームによる描画をサポートすること.	本オプションを選択することで, テクスチャの代わりにワイヤフレームで描画を行う.
5	画像出力	シミュレートした画像は FITS, JPEG, PNG で出力できること.	FITS は惑星探査でも良く使われる形式の 1 つであるため, FITS 形式を出力として含める.
		FITS 形式の場合, WCS ヘッダを含めること.	WCS ヘッダに対応することで, ds9 等の FITS ビューワで確認した際に, 観測領域の赤経赤緯が表示される.
		FITS 形式の場合, カラー出力に対応すること.	ds9 で読み込むことができるカラーフォーマット FITS RGB [24] をサポートすることにより, カラーテクスチャを取り扱うことができる.

config file のフォーマットは, Lua[22] を用いたスクリプト形式を採用している. Lua はスクリプト言語であるが, FLOW のように C 言語で記述されたプログラムから呼び出すことが可能で, Lua スクリプト内で定義した定数は C 言語から参照することができる. また Lua は連想配列に相当するテーブル型を利用することができ, テーブル自身を入れ子構造とすることができる. config file 内では, SPICE カーネルのパス設定や, 描画天体に関連したファイルのパスを設定するために, Lua のテーブルを利用している.

```

-----
--! @file color rosetta.lua
--! @brief FLOW configuration example (ROSETTA 3rd Earth flyby)
--! @copyright Copyright 2010-2012 Japan Aerospace Exploration Agency
--
-- ROSETTA 3rd Earth flyby
--
-- Overview
--
-- This example creates a ROSETTA WAC simulation observed
-- at 2009-11-13T14:23:00. Refer to 'Rosetta Earth swingby successful' to
-- compare.
--
-- Reference
-- Rosetta Earth swingby successful
-- http://www.planetary.org/blog/article/00002212/
-----

-- Kernel directories
generic_kernel_dir = "kernels/generic_kernels/"
rosetta_kernel_dir = "kernels/ROSETTA/kernels/"

-- SPICE kernels
spice_kernels = {
  generic_kernel_dir .. "lsk/naif0010.tls",
  generic_kernel_dir .. "pck/pck00010.tpc",
  generic_kernel_dir .. "spk/planets/de421.bsp",
  rosetta_kernel_dir .. "fk/ROS_V18.TF",
  rosetta_kernel_dir .. "sclk/ROS_110405_STEP.TSC",
  rosetta_kernel_dir .. "ik/ROS_OSIRIS_V11.TI",
  rosetta_kernel_dir .. "spk/ORGR_00096.BSP",
  rosetta_kernel_dir .. "ck/ATPR_P091108000000_00086.BC",
}

-- star data
star_kernel = generic_kernel_dir .. "stars/hipparcos.bdb"
star_magnitude_limit = 4.0

-- view settings.
title = "Sample view from ROSETTA"
time = "2009-11-13T14:23:00"
basis_object = "ROSETTA"
basis_instrument_id = -226112 -- WAC
aberration_correction = "LT+S"

-- object texture and model
texture_file_dir = "texture/"
object_table = {
  [{"PLANET.EARTH"}] = {
    {
      type = "texture-body",
      file = texture_file_dir .. "earth.jpg",
    },
  },
}

-- output objects
output_objects = {
  "SUN",
  "PLANETS",
  "STARS",
}

image_width = 1024
image_height = 1024

```

図2 config file の例。Lua スクリプト形式となっている。

表 2 FLOW のアーキテクチャ対応状況. FLOW は POSIX に対応している OS をサポート. 環境が用意できず検証していない OS は未検証と表記.

Architecture	SPICE Toolkit	FLOW	備考
Mac/Intel, OSX, Apple C, 32bit/64bit	○	○	未検証
Mac/PowerPC, OSX, Apple C, 32bit	○	△	
PC, CYGWIN, gCC, 32bit	○	○	
PC, Linux, gCC, 32bit/64bit	○	○	未サポート
PC, Windows, Microsoft Visual C, 32bit/64bit	○	×	
Sun/Intel, Solaris, Sun C, 32bit/64bit	○	○	
Sun/SPARC, Solaris, Sun C, 32bit/64bit	○	△	未検証

4.3 シーンファイル

シミュレーション画像に恒星などの天体が含まれていた場合、その画像から含まれる天体の名称を判別するのは困難である。そこでシミュレーション過程で得られた描画対象の天体に対して、天体名称や色などの天体情報や、その天体が画像上のどの位置に表示されているのかを示す必要がある。

そこで FLOW では、これらの天体情報を、入力に用いた撮像条件と共に、シーンファイル内に含めて出力する。図 3 はシーンファイルの実サンプルであり、各々のタグの意味を表 3 に示す。シーンファイルには、カテゴリとして描画条件、観測条件、天体情報の 3 つの内容が含まれている。描画条件は、入力となる config

表 3 シーンファイルに含まれる情報. 描画条件および観測条件は 1 シーンファイルに 1 つ存在する. 天体は描画する対象の個数に応じて含まれる.

カテゴリ	タグ	出力項目
描画条件	title	タイトル (FITS を指定した場合にヘッダに追記される)
	image_size	画像サイズ
	sky_color	宇宙空間となる背景色
観測条件	location	観測点位置 (x,y,z)
	boresight	観測機器座標系における視線方向ベクトル (x,y,z)
	bounds	観測機器座標系における視野境界ベクトル (x,y,z) × 個数 (四角形の場合 4 つ)
	fov	視野角 (degrees)
	pos_angle	画面中心から画面上端中央までの角度 (degrees)
	angle_res	画面中心から画面上端中央までのピクセル解像度 (degrees/pixel)
	shape	視野形状 (四角形, 多角形, 円, 楕円)
	center	視野中心の赤経・赤緯
天体情報	type	天体種別 (太陽系天体, 恒星)
	name	名称
	magnitude	明るさ (強度)
	distance	距離
	radius	半径 (天体が点でなく大きさを持つ場合)
	rotation	回転行列 (天体が点でなく大きさを持つ場合)
	spectral	スペクトル (天体が恒星の場合)
	texture	テクスチャ情報 (天体のテクスチャが存在する場合)
	model	ポリゴンモデル情報 (天体のポリゴンモデルが存在する場合)
	image_pos	天体の画像上での位置 (ピクセルの座標)

file に書かれた情報をそのまま出力する。観測条件は、主に SPICE カーネルから読み込んだ情報を出力する。天体情報は、視野角内に含まれる天体の情報を計算し、それら計算値と SPICE カーネルに含まれる天体情報の値を出力する。天体が複数ある場合、シーンファイルには天体情報が複数含まれている。

4.4 描画コンテンツ

4.4.1 テクスチャ画像の張り付け

FLOW は、描画対象となる惑星・衛星が、点ではなく大きさを持つ場合には、ユーザが予め用意したテクスチャ画像を貼付けられる機能を持つ。この機能により、実際の観測とシミュレーションとの比較を容易にすることができる。例えば白地図や経緯度線のみ含まれるテクスチャ画像を用意することで、観測領域の明確化が可能となる。

図 4 は、2007 年 11 月 13 日 19 時 30 分 00 秒 (UTC) に撮像した Rosetta OSIRIS WAC のシミュレーション例である。実際には夜間にあたるため、陰影効果の除去を行い、地表面のテクスチャの明確化を行っている。

用意すべきテクスチャ画像は、縦横比 1:2 かつサイズが 2 のベキ乗となっている必要がある。また左端が経度 0 度、右端が経度 360 度、上端が緯度 90 度、下端が緯度-90 度の画像を用意する (図 4 左下参考)。

画像の解像度は任意であるが、描画領域に比べて用意した画像の解像度が低いと、ズームアップした場合に十分な解像度が得られない。このような場合、全球に渡って高解像度のデータを用意するのは難しく、また用意できたとしても、ファイルそのものが巨大になりソフトウェアの動作に支障を来す恐れがある。

これを解決するために、FLOW は、部分的に高解像度を持つテクスチャが用意された場合に、その部分のみ高解像度に表示することができる。図 5 は、SELENE HDTV WIDE カメラによる 2007 年 11 月 7 日 05 時 56 分 00 秒 (UTC) の撮像をシミュレートしたものである。ベースマップとなる Clementine の画像の上に、SELENE LALT で作成した高度データ (カラー) を経度 170-190 度、緯度 70-90 度の部分のみ重ね合わせている。重ね合わせの効果を分かり易くするために、陰影の効果は除去している。FLOW はさらに、部分的なテクスチャを準備する方法として、Web Map Service (WMS)[26] もサポートし、画像のファイルパスの代わりに、WMS の URL とレイヤータイトルを使用することで、部分的な画像を直接描画することができる。図 6 は、部分テクスチャ画像として WMS を設定するための、config file の一部である。テクスチャ定義を並列に記述することにより、先に記述した内容は、後ろに記述されたテクスチャ定義によって上書きされる。テクスチャが部分的な領域の場合には、部分的に上書きされる。

4.4.2 リングを持つ天体の描画

FLOW は、木星や土星などのリングを持つ惑星を描画する機能を備えている。図 7 は Cassini ISS NAC で 2004 年 3 月 27 日 13 時 10 分 00 秒 (UTC) に撮像した土星の、実画像とシミュレーションの比較である。

リングの表示には、土星のテクスチャ画像の他に、リング用のテクスチャ画像を用意している (図 8)。リングのテクスチャ画像には透明色を利用するため、PNG フォーマットで用意する必要がある。

config file では、texture-body に加えて texture-ring の設定を追加する (図 9)。使用するリングのテクスチャ画像が、木星の半径の何倍に相当するかを rate として与えなければならない。図 9 の例では、木星の外環のリングが、木星半径の 2.31 倍となる画像を用いている。

土星の夜側の陰影の様子から、土星のシミュレーションはうまく機能していることが分かるが、FLOW ではレイトレーシングのように光の道筋を正確にトレースしていないため、シミュレーションの画像 (図 7 右) において、リングの上に土星の影が落ちていないことが分かる。

```

<?xml version="1.0" encoding="UTF-8"?>
<svdoc>
  <head>
    <title>Sample view from ROSETTA</title>
  </head>
  <frame>
    <view>
      <date>2009-11-13T14:23:00.000</date>
      <location>
        <x>-138408685.101606</x>
        <y>-85904.111791</y>
        <z>52053691.578269</z>
      </location>
      <boresight>
        <x>0.000000</x>
        <y>0.000000</y>
        <z>140.000000</z>
      </boresight>
      <bounds>
        <point>
          <x>14.554687</x>
          <y>14.554687</y>
          <z>138.478598</z>
        </point>
        <point>
          <x>-14.554687</x>
          <y>14.554687</y>
          <z>138.478598</z>
        </point>
        <point>
          <x>-14.554687</x>
          <y>-14.554687</y>
          <z>138.478598</z>
        </point>
        <point>
          <x>14.554687</x>
          <y>-14.554687</y>
          <z>138.478598</z>
        </point>
      </bounds>
      <fov>16.909067</fov>
      <pos_angle>327.509055</pos_angle>
      <angle_res>1.171875e-02</angle_res>
      <shape>RECTANGLE</shape>
      <center>
        <ra>352.706650</ra>
        <dec>-24.638810</dec>
      </center>
      <image_size>
        <width>1024</width>
        <height>1024</height>
      </image_size>
      <sky_color>
        <red>0.000000</red>
        <green>0.000000</green>
        <blue>0.000000</blue>
        <alpha>1.000000</alpha>
      </sky_color>
      <object type="solar" id="PLANET.EARTH">
        <name>EARTH</name>
        <position>
          <x>-1828.232707</x>
          <y>-355.078269</y>
          <z>235646.330177</z>
        </position>
        <magnitude>-17.00</magnitude>
        <distance>235653.609641</distance>
        <texture component="body">
          <src type="file">texture/grid.png</src>
        </texture>
        <image_pos>
          <x>474</x>
          <y>534</y>
        </image_pos>
        <radius>
          <x>6378.1366</x>
          <y>6378.1366</y>
          <z>6356.7519</z>
        </radius>
        <rotation>
          <A3>0.8171123481, -0.3059782862, -0.4885741490</A3>
          <A3>-0.5693960767, -0.2959306158, -0.7669505711</A3>
          </A3>
          <A3>0.0900861726, 0.9048769856, -0.4160313960</A3>
        </rotation>
      </object>
      <object type="star" id="STAR.HIPPARCOS.114341">
        <name>114341</name>
        <position>
          <x>0.641700</x>
          <y>-0.096234</y>
          <z>0.994485</z>
        </position>
        <magnitude>3.60</magnitude>
        <distance>2.204055e+15</distance>
        <image_pos>
          <x>716</x>
          <y>40</y>
        </image_pos>
        <spectral>K1</spectral>
        <color>255,160,160</color>
      </object>
      <object type="star" id="STAR.HIPPARCOS.115438">
        <name>115438</name>
        <position>
          <x>-0.015239</x>
          <y>-0.083835</y>
          <z>0.996363</z>
        </position>
        <magnitude>3.96</magnitude>
        <distance>1.535163e+15</distance>
        <image_pos>
          <x>437</x>
          <y>102</y>
        </image_pos>
        <spectral>K0</spectral>
        <color>255,160,160</color>
      </object>
    </view>
  </frame>
</svdoc>

```

図3 シーンファイルの例。シーンファイルはXML形式で出力される。この例では、シミュレーションの結果として、視野角内に含まれる3つの天体情報が出力されている(着色部)。

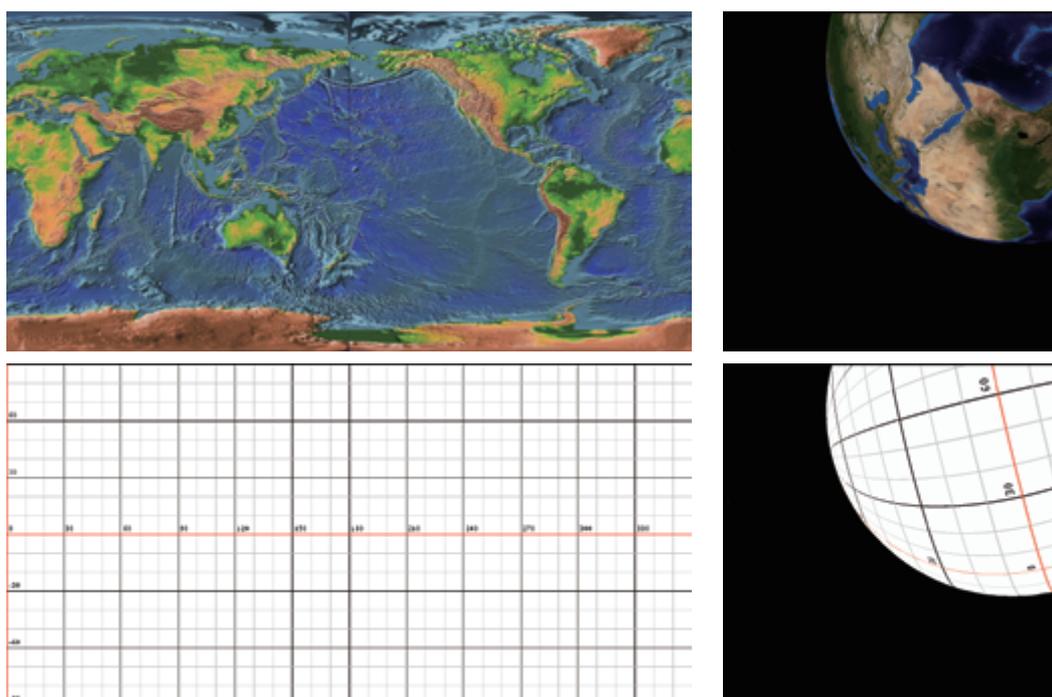


図4 テクスチャ表示例. シミュレーション対象の機器として Rosetta OSIRIS WAC を選択. 地球の地形テクスチャ (上段) と経緯度線グリッドテクスチャ (下段) を用いた. 双方とも左側は使用したテクスチャ画像, 右側はシミュレート結果である.

4.4.3 形状モデルの描画

視野のシミュレートを行う際, 単なる球面にテクスチャを張るだけでなく, 予め用意された形状モデルを描画する機能が必要である. 小惑星探査機はやぶさには, AMICA と呼ばれるカメラが搭載され, 小惑星 25143 Itokawa の観測を行った. これをシミュレートしたのが図 10 である. 形状モデルは小惑星 Itokawa の Gaskell 形状モデル [12] を用いた.

形状データの座標系は, +X 方向が子午線方向, +Z 方向が北極とし, +Y 方向は +X と +Z から右手系となるように決定する. 形状データの中で用いる数値 (距離) の単位は km である. 形状モデルと SPICE の座標系 (FRAME) とのマッピングは, flow_se の設定ファイルである config file で行う.

FLOW では, 形状モデルに対してテクスチャを張る機能を有しておらず, 単色の物体として, config file 内に形状モデルの色を RGB で設定する項目を含んでいる. 例えば小惑星 Itokawa のシミュレーションを行う場合には, (Red,Green,Blue)=(204,204,204) という値を config file に設定している.

4.4.4 視野形状対応 (多角形・四角形・円・楕円)

SPICE Toolkit では, “POLYGON”, “RECTANGLE”, “CIRCLE”, “ELLIPSE” の 4 つの視野形状に対応している. FLOW も同様に, これら 4 つの視野角を取り扱うことができる. FLOW における視野形状の処理は, 最初に視野全体を含む長方形を計算し, 次に視野以外の部分をマスクする方法で実現している.

図 12 は, Venus Express に搭載されたカメラ VMC による実画像とシミュレーション画像の比較である. 対象として 2014 年 01 月 07 日 01 時 04 分 59 秒 (UTC) に撮像された画像を用いた. VMC の視野は八角形の形状をしており, 多角形形状である視野がマスクされていることが分かる. そのままシミュレーションを行うと, マスク部分が黒のために, 視野の境界が明確とならない場合がある. そこで config file 上でバックグラ

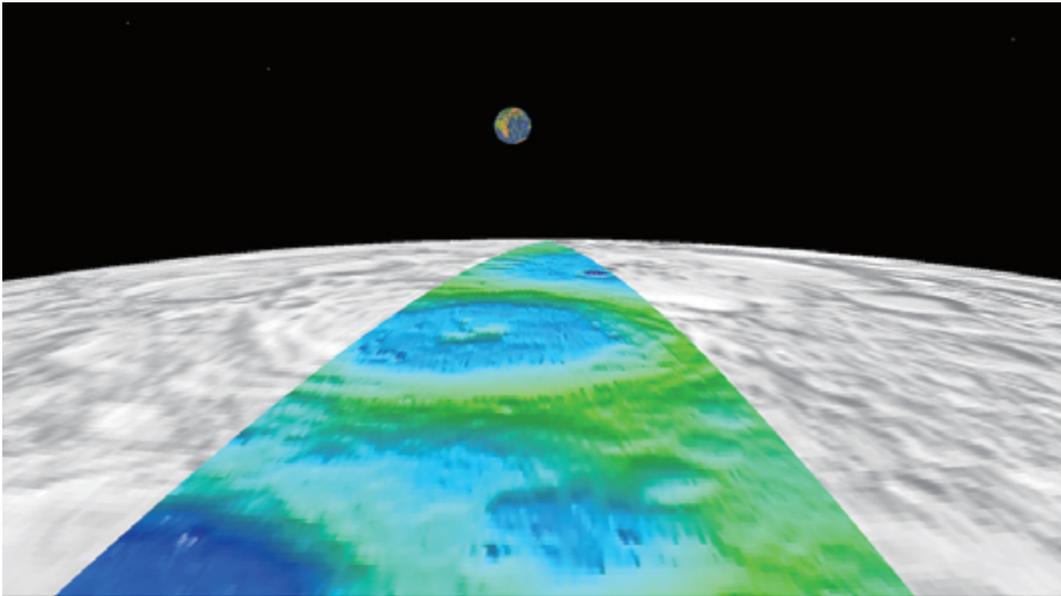


図5 FLOWの部分テクスチャ機能例。2007年11月07日05:56:00(UTC)のSELENE HDTV WIDEカメラ撮像をシミュレート。Clementineの全球テクスチャにSELENEレーザー高度計(LALT)によるテクスチャの重ね合わせている。重ね合わせの効果が分かりやすいよう、陰影の効果は除去している。

```
[ "SATELLITE_MOON" ] = {
  {
    type = "texture-body",
    file = texture_file_dir .. "moon.png",
  },
  {
    type = "texture-body",
    wms_url = "http://onmoon.jpl.nasa.gov/wms.cgi",
    wms_title = "LRO Colorized and Shaded Elevation, second version",
    region = "170.0,70.0,190.0,90.0",
    resolution = 50
  },
},
```

図6 部分テクスチャ画像による上書きを行う config file 例。ここでは WMS サーバを使用して、部分テクスチャ画像をプログラム実行時に読み込んでいる。

ウンド色 (sky.color) として深い青色を指定してシミュレーションを行っている。

4.4.5 プッシュ・ブルーム対応

周回衛星などに搭載されるカメラには、衛星の移動方向と垂直に設置した1次元センサーによってスキャンしていくことで2次元画像を作る「プッシュ・ブルーム方式」と呼ばれるカメラがある(図13)。SPICEの視野定義としては横4096ピクセル、縦1ピクセルといった定義がなされる。これをそのまま描画すると、横に長い画像が作成され、意図したシミュレーション画像が得られない。そこで本ソフトウェアでは観測時刻の代わりに、観測開始・終了時刻と時間刻み幅が与えられた場合には、プッシュ・ブルーム方式として動作する機能を追加している(図14)。

FLOWは3軸固定の周回衛星がラインセンサを使用した際のシミュレーションに対応しているが、1軸スピナーの衛星がスキャンして作成する画像については、今のところ対応していない。

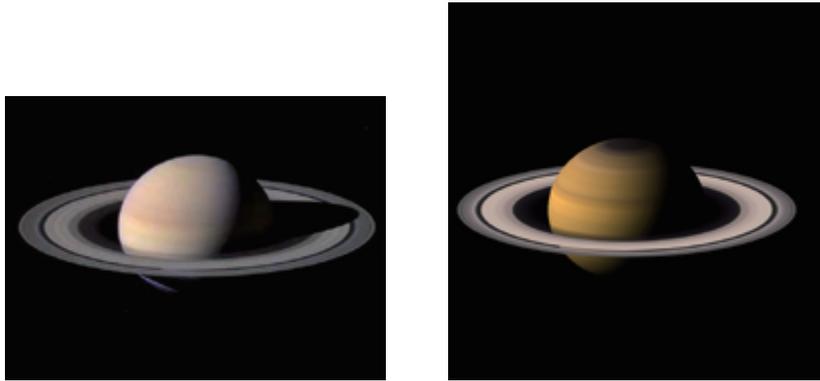


図7 土星リングの表示例. 2004年03月27日13:10:00(UTC)のCassini ISS NACを対象としてシミュレート. 実際の観測画像(左)とFLOWによるシミュレーション画像(右).

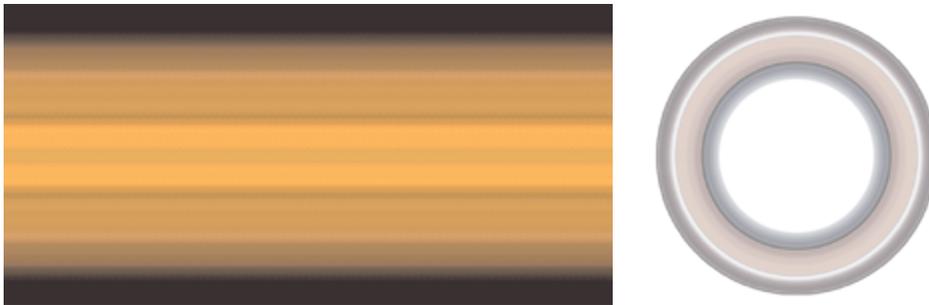


図8 土星表示に使用するテクスチャ画像. 土星表面のテクスチャ画像(左)とリングのテクスチャ画像(右). リング用のテクスチャ画像はPNGフォーマットによる透明部分を含んでいる.

```
object_table = {
  ["PLANET.SATURN"] = {
    {
      type = "texture-body",
      file = texture_file_dir .. "saturn.png",
    },
    {
      type = "texture-ring",
      file = texture_file_dir .. "saturn_ring.png",
      radius = "2.31",
    },
  },
},
}
```

図9 土星の環を表示するためのconfig file内の設定例. texture-bodyに加えて, texture-ringの設定を追加する.

4.5 描画オプション

描画オプションとは, 実際の描画を担うflow.igに対するオプションである. シミュレーション条件や描画コンテンツとは独立に用意された機能であり, どのシミュレーションにおいても描画時に指定することができる.

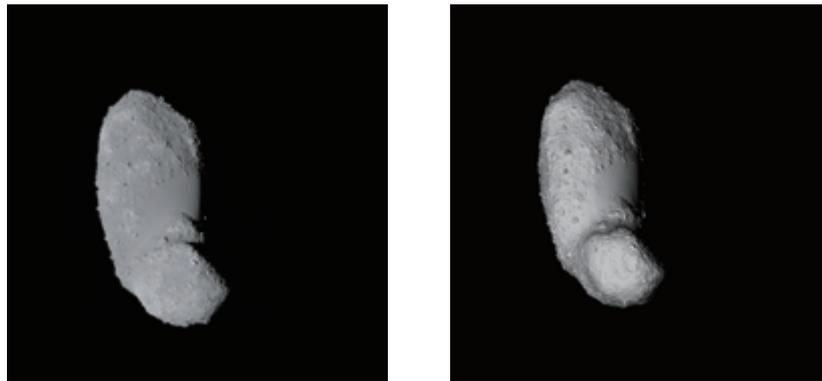


図 10 形状モデルシミュレーション例. 2005 年 10 月 16 日 13:30:13(UTC) に取得した Hayabusa AMICA による実際の観測画像 ST_2465140661_v(左) と FLOW によるシミュレーション画像(右). 形状モデルの色は, config file 内で灰色に設定している.

```
object_table = {
  ["ITOKAWA"] = {
    {type = "model",
     file = model_file_dir .. "itokawa_f0049152.stl",
     scale = 1.0,
     color = "204,204,204",
     format = "STL"}
  }
}
```

図 11 ポリゴンモデルを表示するための config file 設定例.

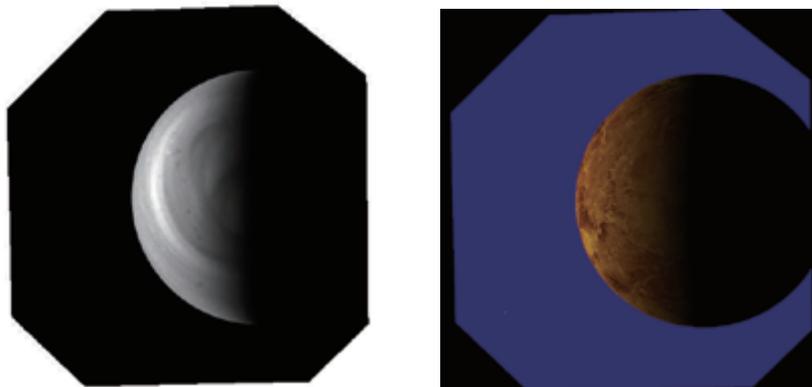


図 12 多角形視野のシミュレーション例. Venus Express UV が 2014 年 01 月 07 日 01:04:59(UTC) に行った金星観測をシミュレートした. 左は実際の観測画像, 右は本ソフトウェアによるシミュレーション画像. 多角形の視野が分かりやすいよう, sky_color オプションで背景に深い青色を用いている.

4.5.1 陰影効果の除去

図 15 は, 陰影の効果除去した例である. 通常のシミュレーション時には, 陰影の効果によって月面が半分だけ見えている. 陰影の効果除去すると, 月面が全面見えるようになる.

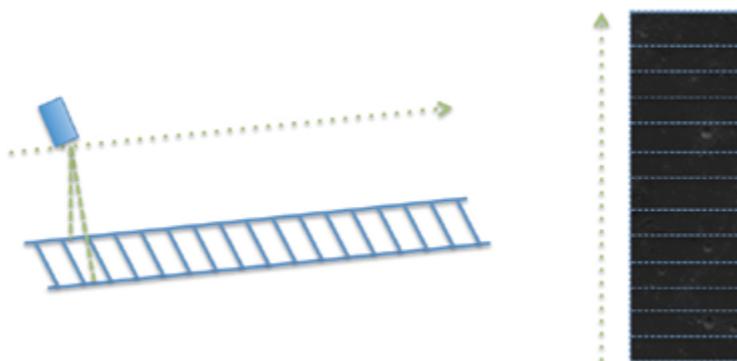


図13 プッシュ・ブroom方式による撮像. 2008年05月20日21:26:00から10秒間, SELENE TCの撮像をシミュレート. ラインセンサを衛星の移動方向と垂直に置くことで, 二次元イメージを作成することができる.

```
time_start = "2008-05-20T21:26:00"
time_end   = "2008-05-20T21:26:10"
time_step  = 0.0075
```

図14 プッシュ・ブroom方式で描画するための config file 設定例.



図15 陰影効果を無視したシミュレーション例. 2007年11月14日06:36:00(UTC)に実施したRosetta OSIRIS NACによる地球スイングバイ後の月観測を模擬している. 陰影あり(左)と陰影なし(右)をそれぞれシミュレートしている.

4.5.2 ワイヤーフレーム

ワイヤーフレームは, 対象となる天体にテクスチャを張らずに, それらをワイヤーで接続した描画を行う. 図16は, ワイヤーフレームの効果を表した例である. SELENE HDTVの望遠カメラ(TELE)が, 2008年04月05日21時44分26秒(UTC)に月面から見た地球の夜明けをシミュレートしたものである. 図16左は通常のシミュレーションに陰影の効果を除いたものである. 図16右は, ワイヤーフレームオプションをつけ, かつ陰影の効果を除いたものである. 月面および地球がワイヤーフレームとなっていることが分かる.

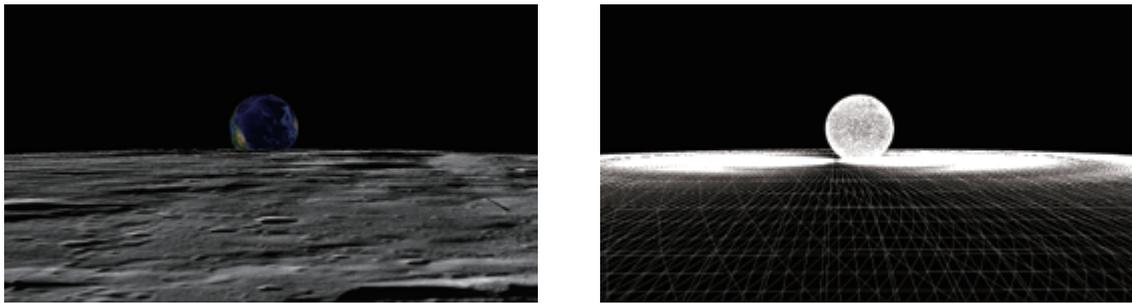


図 16 ワイヤーフレームシミュレーションの例. 2008 年 04 月 05 日 21:44:26(UTC) を使用. SELENE HDTV のシミュレーション (左) とワイヤーフレームシミュレーション (右). 双方とも陰影の効果は除いている.

4.6 画像の出力形式

画像を出力する上で、一般的な画像形式に対応することは、様々なシステムに組み込む上で重要な点である。FLOW では、一般的なフォーマットとして JPEG および PNG による出力をサポートしている。

一方で、天文分野で標準として使用され、近年では惑星探査においてもしばしば利用される FITS フォーマットもサポートしている。FITS による出力では、ファイル自身のヘッダ部に World Coordinate System (WCS)[14] 固有のキーワードを含んでいる。そのため ds9 [25] 等の WCS に対応した FITS ビューワで出力された画像を開いた場合に、赤経・赤緯が表示される (図 17)。

5 ソフトウェアの実装

5.1 恒星の表現

恒星の表現は、AstroArts 社 [1] で開発したステラナビゲータに由来した、視覚・体験に基づいた表現を行っている。恒星のサイズや明度は、明るさに応じて、8 段階に変化させている (図 18)。またスペクトル型に対応して色表現を変えている (図 19, 表 4)。スペクトル型の情報は、SPICE カーネルの恒星データベース (例えば Hipparcos カタログ) の情報を参照する。

表 4 恒星のスペクトル型と RGB 対応

恒星のスペクトル型	(Red,Green,Blue)
O	(128,128,255)
B	(160,160,255)
A	(192,192,255)
F	(224,224,255)
G	(255,192,192)
K	(255,160,160)
M	(255,128,128)

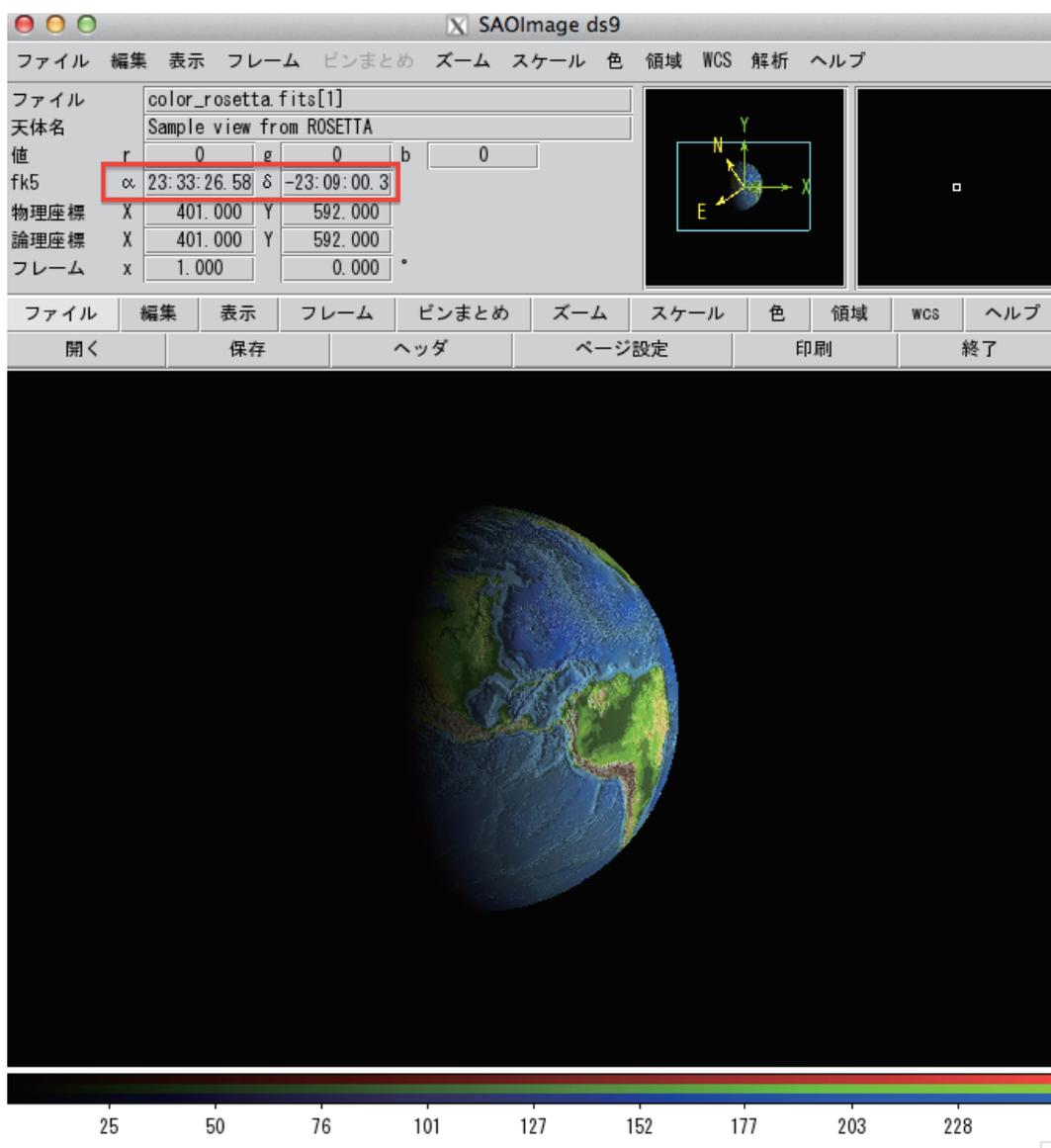


図 17 ds9 による出力結果の表示. シミュレーションには Roseta OSIRIS WAC が 2009 年 11 月 13 日 14:23:00(UTC) に撮像した地球スイングバイ時の時間を選択した. FITS RGB/FITS cube によるカラー対応と, WCS ヘッダによる赤経・赤緯の表示に対応している (赤枠部分).

5.2 独自の描画エンジン

より多くのプラットフォームで動作させるために, FLOW は, それ単体で動作する独自の描画エンジンを搭載している. 一つは 2D グラフィックエンジン (2D-canvas) であり, もう一つは 3D グラフィックエンジン (Tiny 3D Graphic Engine; 3D-tge) である.

2D-canvas は, オンメモリで動作する低レベルルーチンであり, FITS/JPEG/PNG の入出力となる Image I/O と接続する (図 20 左). ピクセルのカラー指定, カラーブレンド, canvas 同士の掛け合わせに対応している.

3D-tge は, ステラナビゲータ用に 2000 年に開発された技術を土台としている. 三角形ポリゴンの描画機能を有し, 陰影処理, 深度バッファ, アルファブレンド機能に対応している. 3D-tge の描画は, 2D-canvas を



図 18 恒星の明るさに応じたサイズ・明度の変化. 恒星の明るさは 8 段階で表現を行い, 左端が Magnitude 7.5~6.5 に対応し, 右に行くに連れ Magnitude が 1.0 下がる. 右端は Magnitude 0.5 未満に対応する.



図 19 スペクトル型による色表現の変化. 左から恒星のスペクトル型 O, B, A, F, G, K, M に対応する.

バックエンドとして利用する (図 20 右).

ハードウェアレンダリングを必要としないため, OpenGL 不要, X-Window System 不要, GPU 非依存であり, 移植性の高いソフトウェアを実現している. また高速化要求のために OpenGL に移行しやすいよう, 3D-tge の関数名の命名規約は OpenGL と似た関数名を採用している. 例えば OpenGL で RGBa カラー形式で色を指定する場合に用いる関数 `glColor4f()` は, 3D-tge では `tgeColor4f()` に対応している.

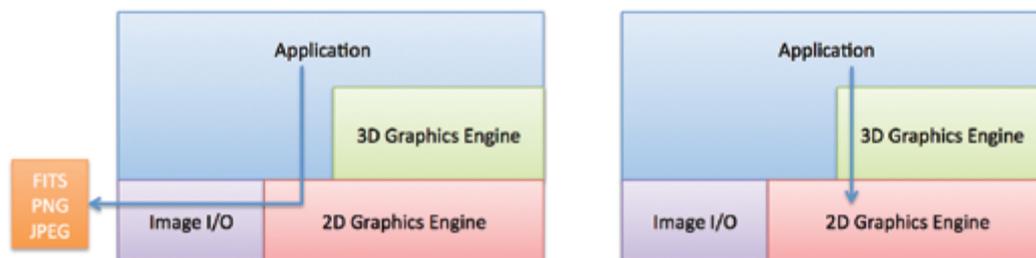


図 20 搭載グラフィックス・エンジンの構造. 2D グラフィックスは FITS/PNG/JPEG の Image I/O と接続する (左). 3D グラフィックスは 2D グラフィックスをバックエンドとして使用 (右).

5.3 形状モデルの読み込み

形状モデルのフォーマットは, 一般的なフォーマットである STL(STereoLithography) フォーマットと, SPICE Toolkit のアルファ版で提供している SPICE DSK Type 2 に対応している.

STL フォーマットの構文解析については, ステートマシンコンパイラ Ragel[28] を利用して C 言語のソースコードを出力し, これを組み込んでいる.

SPICE DSK Type 2 は C 言語による API が提供されているため, これを用いて形状データを読み出している. NAIF による SPICE DSK Type 2 の開発状況は現在 α 版であるが, 小惑星 25143 Itokawa, 小惑星 2867 Stein, 小惑星 4 Vesta, 火星の衛星 Phobos が公開されており, FLOW ではこれらを表示することができる (図 21).

5.4 依存ライブラリとライセンス

本ソフトウェアは, SPICE 同様, 一般に広く利用してもらうことを前提に開発を行い, 修正 BSD ライセンス [19] で公開している. また本ソフトウェアを利用したいユーザが, その使用に対して制約がないよう,

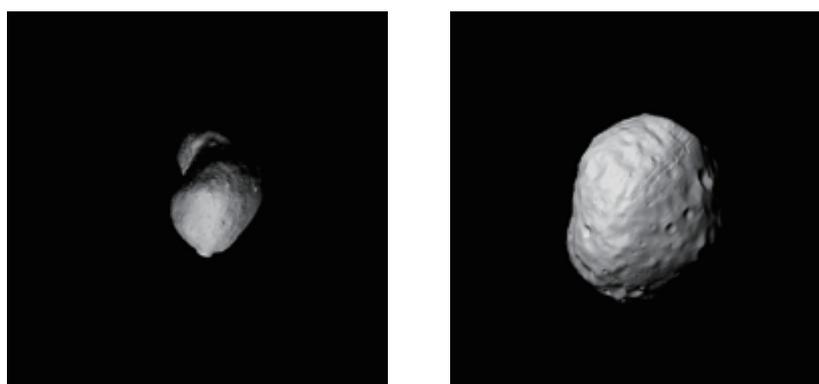


図 21 形状モデルの読み込み。左は STL フォーマットによる小惑星 25143 Itokawa(撮像時刻 2005 年 10 月 10 日 07:35:40 UTC)。右は SPICE DSK Type 2 フォーマットによる火星の衛星 Phobos(撮像時刻 2006 年 01 月 20 日 23:07:49 UTC)。

ソースコード公開の義務のある GPL[5] や LGPL[6] を避けている。また商用でも利用可能なように、Public domain, 修正 BSD ライセンス, MIT ライセンスを主体としたソフトウェア構成としている (表 5)。

しかしながら、実際に本ツールの派生ソフトウェア等を開発したいと考えた場合には、次の 2 つの点について注意が必要である。一つは libxml2 がリンク可能な iconv2 のライセンス、もう一つは libgcc に含まれる libregex である。これら 2 つのソフトウェアは共に LGPL を採用しているため、本ツールの派生ソフトウェアを開発しソースコードを非公開のまま再配布したいと考える場合には、libxml2 から iconv2 を除き、かつ libregex を libgcc の一部として利用するなど工夫が必要である。

表 5 依存ライブラリのライセンス

ライセンス	依存ライブラリ	備考
MIT	libxml2, Lua, libcurl	libxml2 は標準で iconv2 をリンク
Original	SPICE Toolkit, cfitsio, libpng, jpeglib	
LGPL	iconv2, libregex	派生ソフトウェア配布時に注意が必要

5.5 パッケージング

SPICE Toolkit では、複数のプラットフォームで動作するように、プラットフォームごとにソースコードを分けている。FLOW では、libxml2 や libpng など、複数の既存ライブラリに依存しているため、ソースコード配布時に頻繁に利用されている Autoconf[3] および Automake[4] を使用して各種プラットフォームに対応している。これによりソースコードを修正することなく、マルチプラットフォームでコンパイルを実現できる設計としている。また複数のプラットフォームでのコンパイル例を実例として示すことにより、コンパイルで迷う手間を低減している。

6 一般ユーザ向けのインターフェイス

6.1 一般ユーザ対応

本シミュレータは、利用者として SPICE の習得度についてある程度のレベルを求めている。少なくとも視野角を計算するために必要な SPICE カーネル群の準備や、対象とする観測機器に割り当てられている NAIF

ID についての知識が必要である。実際にこれが可能なのは、観測機器チームであったり、過去に惑星探査に携わり SPICE を利用した経験があるユーザである。本セクションでは、SPICE の修得を隠蔽し、一般向けのユーザのために開発したインターフェイスについて記述する。

6.2 開発コンセプトと開発要件

一般ユーザが利用する場合、観測機器の選択と時刻があれば、シミュレーションに必要な情報は十分である。また描画時のオプションとして、陰影の有無、テクスチャの選択ができれば、FLOW が持つ機能を損なうことなく、ユーザが求める描画に対応することができる。しかし一般ユーザの中には、実際にどのような時刻を入れて良いのか分からない場合があるため、これをサポートする仕組みも必要である。加えて、恒星や惑星が画像に含まれていた場合に、どの天体であるのかを識別する必要がある。これらを実現するために、以下の要求を開発要件とした。

1. ユーザが時刻を入力し、送信ボタンを押した場合に、シミュレートした画像を表示すること
2. 観測機器が複数存在する場合、観測機器を選択できること
3. 画像の表示オプションとして、陰影あり・なし、ワイヤーフレームによる表示、高解像度テクスチャの表示を選択できること
4. ユーザの入力をサポートするために、具体的な例をマウスで選択した場合に、時刻の入力・観測機器の選択をサポートすること
5. 画像の描画に時間がかかる場合、待ち状態であることを分かるような表示を行うこと
6. 画像に表示された天体が分かるよう、天体の情報と画像上の位置を表示すること
7. 画像の現在位置を知るために、マウスオーバー時にカーソルポジションを表示すること

6.3 ウェブブラウザベース・アプリケーション

より多くのユーザに対応するために、一般ユーザ向けのインターフェイスとして、ウェブブラウザベースのアプリケーションを選択した (図 22)。これを FLOW on the web と名付けインターネット上で公開している [11]。デスクトップユーザだけでなく、モバイル等でアクセスするユーザにも対応している。

入力項目は時刻のみ、描画オプションとして、陰影の有無、ワイヤーフレームの選択が行えるようになっている。また高画質のテクスチャやポリゴンモデルが存在する場合には、それらも選択可能としている。

時刻の入力は、ISO 8601 に準拠した YYYY-MM-DDThh:mm:ss という形式を取っている。エディットボックスには、薄くフォーマットが表示される工夫を行っている。また、日付と時刻の間に、アルファベットの T を入れることを一般のユーザが気づかない可能性も考慮し、Date Examples を掲載し、参考となる入力時刻を同形式で表示している。また Date Examples は、クリックすると、JavaScript により時刻と観測機器が自動的に設定されるようになっている。Date Examples の例をクリックし、Draw ボタンを押すだけで画像が描画される。

Date Examples は、各々が Reference に示す参考画像と対応しており、FLOW によるシミュレーションが正常に動作しているかどうかを確認できる時間帯を選んで例示している。実際のシミュレーション結果は右に表示し、画像の上には時間と描画オプションを表示している。画像の下には、視野内に含まれる天体情報を表形式で掲載している。

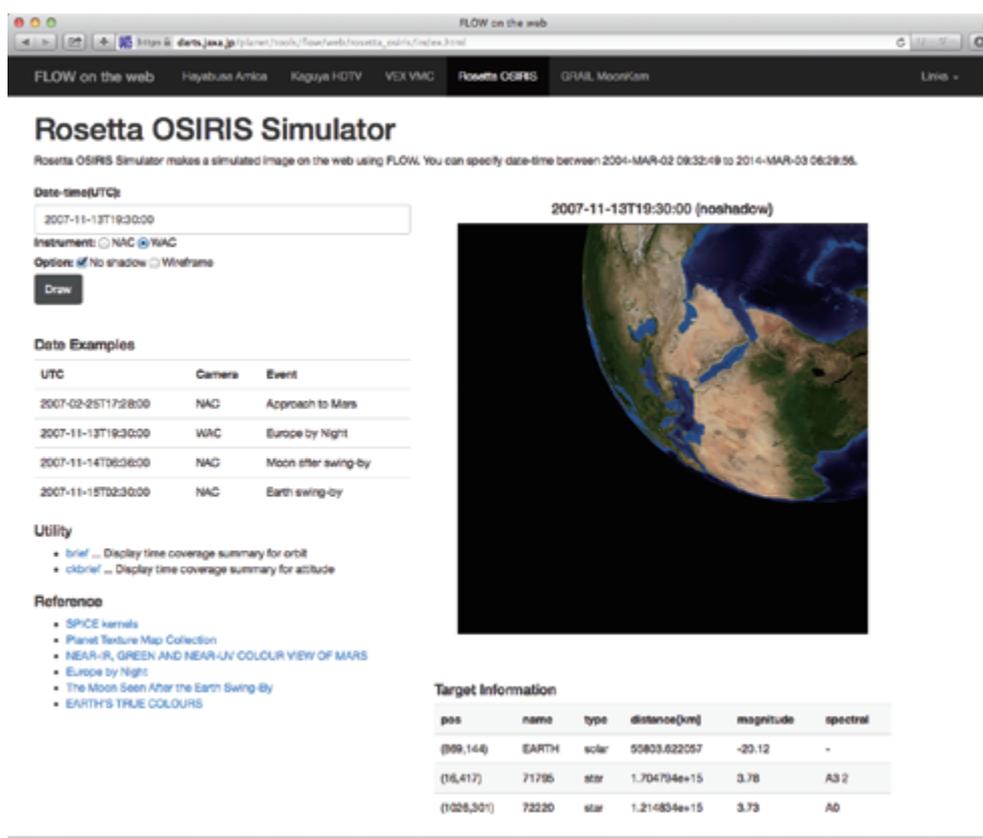


図 22 FLOW on the web のウェブサイト。

6.4 技術的な仕組み

図 23 は、FLOW on the web を模式的に描いた図である。最初にブラウザが HTML ファイルをダウンロードすることから始まる。この HTML ファイルには、JavaScript が含まれており、また関連する JavaScript 群も一緒にウェブサイトからダウンロードする。

ユーザが時刻を入力した後、Draw ボタンを押すと、サーバ上で json.php が実行される。json.php は、主に次の 3 つの処理を行う。

1. テンプレートを読み込み、描画オプションに応じた config file を作成
2. 作成した config file を元に、コンソールプログラム (flow_se と flow_ig) を実行
3. 実行が終了した際に、シーンファイル、画像の URL を JSON 形式 [2] で返却

FLOW のコンソールプログラムが画像を作成するまでに時間がかかるため、AJAX による非同期処理を行っている。非同期処理では、JSON 形式のデータがダウンロードされるまで待ち状態 (アニメーション gif の表示) を続け、データがダウンロードされた後、そこに含まれる URL にアクセスして描画を行う。

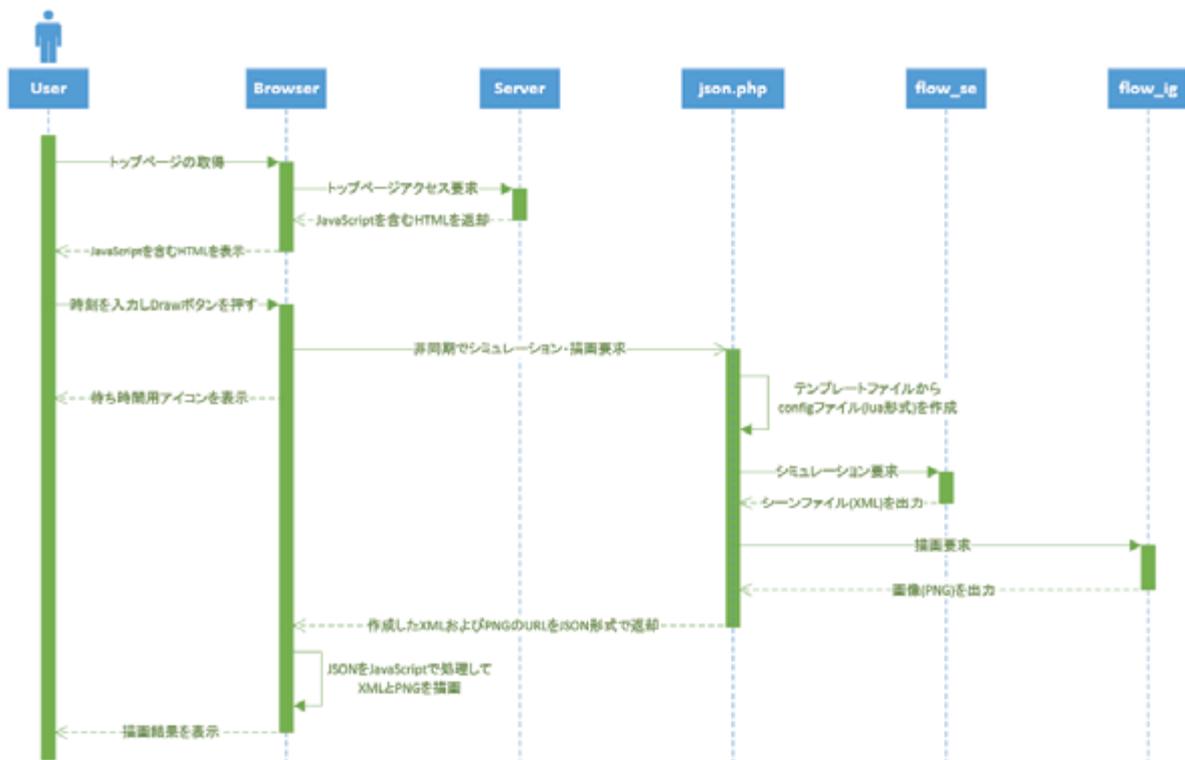


図 23 FLOW on the web の仕組み. コンソールプログラムの FLOW を PHP プログラムから実行する. AJAX により, 描画が終わるまで非同期で待ち, 描画終了と共に JSON が応答として返される. その後 JavaScript でシーンファイルと画像をダウンロードする.

7 発展の方向性

本ソフトウェアの開発は, より汎用的な手法で探査機運用や科学データのパイプライン処理の高次データ作成に利用可能なツールとすることを目的としている. しかし惑星探査において視野角を持つ機器のシミュレーションに対する要求は様々であり, 目的によっては本ソフトウェアでは達成できない項目もある. 本ソフトウェアはオープンソースのソフトウェアであり, ライセンスも修正 BSD ライセンスによって公開しているため, 必要に応じて利用者が改良することが可能である. ここでは本ソフトウェアでは現時点で実現できないシミュレーションについて言及し, 発展の方向性について記述する.

7.1 レンダリング手法

描画の過程において, レイトレーシングを実施していないため, ある物体が別物体に対して光や影の効果を再現することができない. 例えば土星などのリングを持つ惑星がリングに落とす影を再現することはできず(図 7), 同様に地球照のような地球の照り返しをシミュレーションすることはできない. また探査機自身によって作られる影はシミュレーションされない. 図 24 は, 2005 年 11 月 19 日(UTC)に撮像されたはやぶさ搭載の広角カメラ ONC-W1 による実際の撮像と, 三浦らによって POV-Ray[27]を使って可視化された画像[13], さらに FLOW によるシミュレーションの比較である. 公開されている SPICE カーネルには ONC-W1 の視野情報を含んでいないため, 簡易的に AMICA の視野角を 5.7 度から 60 度に変更してシミュレーション

を実施している。これらを比較すると、FLOW のシミュレーションでは探査機の影が実現されていないことが分かる。レイトレーシングによる描画機能の実装は、影が描画に強く影響を及ぼす場合に必要な機能であるが、シミュレーションに必要な時間が増大し、時間制約の限られた探査機運用や地上データ処理に組み込むには不向きであると言える。そのため現段階においてはレイトレーシングによるレンダリングは実装のスコープから外している。

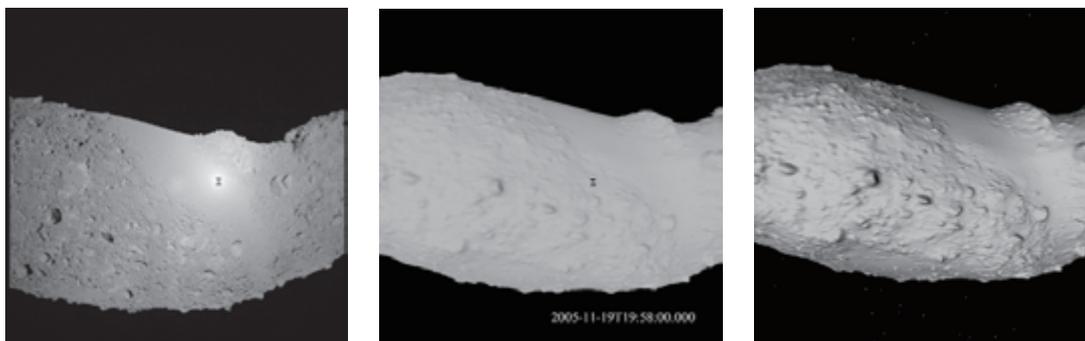


図 24 Itokawa 上に落ちるはやぶさの影。左は実画像，中は三浦らの POV-Ray によるシミュレーション，右は FLOW によるシミュレーション。FLOW によるシミュレーションでは、はやぶさの影は実現できていない。

さらに本ソフトウェアでは反射モデルとして Lambert 反射を用いている。もしそれ以外の反射モデル (例えば Hapke モデル [7] や Phong モデル [21]) を用いたい場合には、3D-tge の反射率計算を改良しなければならない。しかしながら、本ソフトウェアの主目的である探査機運用および地上データ処理の観点から、これらの反射モデルを適用することは現時点で考えていない。

7.2 3次元形状モデルの表現

本ソフトウェアは、小惑星などの3次元形状モデルを描画することができるが、STL と SPICE DSK type 2 のみをサポートしていることから分かる通り、形状モデルの各面に対して色を表現することはできない。現在の FLOW におけるポリゴンは単色であり、設定ファイルに記述する方式を採用している。

NASA は 3D モデルをインターネットで公開しており、ここでは Apollo Lunar Module などダウンロードすることができる [15]。これらのデータは .3ds という色も含めた 3 次元形状を表現可能なフォーマットで公開されている。これらのモデルを扱えるようになることは、月面着陸のシミュレーションや分離機構のある探査機では必要になる可能性がある。

またテクスチャを切り替えられる機能はユーザの理解を助けるために重要な機能であり、形状が球体である月・惑星の表示についてはテクスチャを自由に選択できるようになっている。一方で形状モデルに関してはこの機能を有していない。平田らのチームは、小惑星 Itokawa 上に、単色ではないテクスチャを貼付けるソフトウェアを開発している [8]。将来的にはこの技術や色を含めた形状を表現できるフォーマットを参考に、ポリゴン上へのテクスチャ画像の張り付けを改良することも可能である。

7.3 探査機に関連した改善点

本ソフトウェアは、3軸安定の探査機を考慮して作られている。開発の初期には探査機が動くことにより画像が作られるプッシュ・ブルーム方式には対応していなかった。しかしながらプッシュ・ブルーム方式は周回衛星では比較的良く利用される方式であり、実際に SELENE の地形カメラがそれを意識した SPICE カーネルを作成していたため、機能として追加を行った。同様の理由で、将来的には 1 軸スピナー衛星のスキャン画

像への対応も必要となる可能性がある。

また惑星探査の運用で利用するという目的においては、仮想的な補助線を用いて理解を助けることも考えられる。例えばはやぶさ2の対象天体である小惑星1999JU3は、その形状が球形に近く、シミュレーション結果だけではその方向が分かりにくい。このような場合には、自転軸や赤道、あるいは小惑星固定座標系のXYZ軸が表示される機能が必要となる。その他にも、観測領域に星が写り込んでいる場合は、補助線として星座線を描画することも良いであろう。シーンファイルなしで、直感的にユーザに観測領域を伝えることが可能となる。

8 おわりに

本論文では、SPICEを利用した汎用的な視野可視化のソフトウェアFLOWについて、実際のシミュレーション結果と併せて、機能と実装の詳細について紹介した。

探査機運用および地上データ処理を目的としたソフトウェアとして、様々なプラットフォームで動作させることを念頭に開発を行い、その結果プラットフォームの多様化した運用室の端末にも組み込むことが可能なソフトウェアとなった。さらに地上データ処理で作られるダミーデータとして、過去のミッションデータの流用ではなく、予想軌道や予測姿勢などを考慮したより精度の高いダミーデータの作成を実現できるようになった。

またSPICEがサポートする視野角の形状(多角形, 円, 楕円)を実現でき、さらに探査機が動くことによって画像を作りあげるプッシュブルーム方式にも対応した。今後行われる多くの探査機, 多くの観測機器で利用されることが期待できるソフトウェアとなった。

加えてSPICEを使いこなす専門家だけでなく、FLOWをバックエンドのエンジンとして利用した一般ユーザー向けのウェブサイトの開発も行い、FLOWと連携したアプリケーションの可能性も示した。

一方で、現時点でFLOWでは実現できないシミュレーションについても論じ、将来の改良の方向性を示した。しかしながら、多様化する要求に対して全ての機能を実現することは困難であり、その代わりにソフトウェア自身をオープンソースで公開し、幅広く利用可能な修正BSDライセンスを採用した。この意味は、より多様化した要求を実現するために、要求を持つユーザ自身が本ソフトウェアを成長させてくれることを望んだものである。

参考文献

- [1] ASTROARTS INC. AstroArts - アストローツ. <http://www.astroarts.com/index-j.html>. (参照 2014 年 12 月 1 日).
- [2] BRAY, T. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 7159, RFC Editor, March 2014.
- [3] GNU PROJECT. Autoconf - gnu project - free software foundation (fsf). <http://www.gnu.org/software/autoconf/>. (参照 2014 年 12 月 1 日).
- [4] GNU PROJECT. Automake - gnu project - free software foundation (fsf). <http://www.gnu.org/software/automake/>. (参照 2014 年 12 月 1 日).
- [5] GNU PROJECT. Gnu general public license, version 3. <http://www.gnu.org/licenses/gpl.html>, June 2007. (参照 2014 年 12 月 1 日).
- [6] GNU PROJECT. Gnu lesser general public license, version 3. <http://www.gnu.org/licenses/lgpl.html>, June 2007. (参照 2014 年 12 月 1 日).
- [7] HAPKE, B. W. A theoretical photometric function for the lunar surface. *J. Geophys. Res.*68 (Aug. 1963), 4571–4586.
- [8] HIRATA, N., KAWAMAE, W., TUAN, A., D., KITAZATO, K., DEMURA, H., AND ASADA, N. ポリゴン形状モデルを基盤とした不規則形状小天体観測データ検索・解析システム. 宇宙航空研究開発機構研究開発報告 JAXA-RR-11-007 1 (Mar. 2012), 133–139.
- [9] IEEE STANDARDS ASSOCIATION. IEEE SA - POSIX - Austin Joint Working Group. <http://standards.ieee.org/develop/wg/POSIX.html>. (参照 2014 年 12 月 1 日).
- [10] JAPAN AEROSPACE EXPLORATION AGENCY. Field Of View Visualizer. <http://darts.jaxa.jp/planet/tools/flow/>. (参照 2014 年 12 月 1 日).
- [11] JAPAN AEROSPACE EXPLORATION AGENCY. FLOW on the Web. <http://darts.jaxa.jp/planet/tools/flow/web/>. (参照 2014 年 12 月 1 日).
- [12] JAPAN AEROSPACE EXPLORATION AGENCY. Hayabusa Project Science Data Archive. https://darts.jaxa.jp/planet/project/hayabusa/shape_ja.pl. (参照 2014 年 12 月 1 日).
- [13] MIURA, A., YAMAMOTO, Y., AND YOSHIKAWA, M. 「はやぶさ」の小惑星イトカワ探査軌道・姿勢の可視化. 宇宙航空研究開発機構研究開発報告 JAXA-RR-13-010 3 (Mar. 2014), 7–15.
- [14] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. Fits world coordinate system (wcs). http://fits.gsfc.nasa.gov/fits_wcs.html. (参照 2014 年 12 月 1 日).
- [15] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. Home — 3d resources (beta). <http://nasa3d.arc.nasa.gov>. (参照 2014 年 12 月 1 日).
- [16] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. SPICE. <http://naif.jpl.nasa.gov/>. (参照 2014 年 12 月 1 日).
- [17] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. SPICE DSK. http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/45_shape_model_preview.pdf. (参照 2014 年 12 月 1 日).
- [18] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. Toolkits for C. http://naif.jpl.nasa.gov/naif/toolkit_C.html. (参照 2014 年 12 月 1 日).
- [19] OPEN SOURCE INITIATIVE. The bsd 3-clause license — open source initiative. <http://opensource.org/licenses/BSD-3-Clause>. (参照 2014 年 12 月 1 日).

- [20] PERRYMAN, M. A. C., LINDEGREN, L., KOVALEVSKY, J., HOEG, E., BASTIAN, U., BERNACCA, P. L., CRÉZÉ, M., DONATI, F., GRENON, M., GREWING, M., VAN LEEUWEN, F., VAN DER MAREL, H., MIGNARD, F., MURRAY, C. A., LE POOLE, R. S., SCHRIJVER, H., TURON, C., ARENOU, F., FROESCHLÉ, M., AND PETERSEN, C. S. The HIPPARCOS Catalogue. *A&A* 323 (July 1997), L49–L52.
- [21] PHONG, B.-T. Illumination for Computer Generated Pictures. *Communications of the ACM* 18, 6 (1975), 311–317.
- [22] PONTIFICAL CATHOLIC UNIVERSITY OF RIO DE JANEIRO. The Programming Language Lua. <http://www.lua.org/>. (参照 2014 年 12 月 1 日).
- [23] ROCK, S. J., AND WOZNY, M. J. A flexible file format for solid freeform fabrication. In *Proceedings Solid Freeform Fabrication Symposium* (1991), pp. 1–12.
- [24] SMITHSONIAN ASTROPHYSICAL OBSERVATORY. FITS RGB. <http://zeus.asu.cas.cz/extra/ds9/file.html>. (参照 2014 年 12 月 1 日).
- [25] SMITHSONIAN ASTROPHYSICAL OBSERVATORY. Saoimage ds9. <http://ds9.si.edu/site/Home.html>. (参照 2014 年 12 月 1 日).
- [26] THE OPEN GEOSPATIAL CONSORTIUM (OGC). Web Map Service — OGC. <http://www.opengeospatial.org/standards/wms>. (参照 2014 年 12 月 1 日).
- [27] THE POV-TEAM. POV-Ray - The Persistence of Vision Raytracer. <http://www.povray.org/>. (参照 2014 年 12 月 1 日).
- [28] THURSTON, A. D. Parsing computer languages with an automaton compiled from a single regular expression. In *CIAA 2006: Proceedings of the 11th International Conference on Implementation and Application of Automata* (2006), O. H. Ibarra and H.-C. Yen, Eds., vol. 4094 of *Lecture Notes in Computer Science*, Springer, pp. 285–286.