

SPICEを用いた 視野の可視化と技術選定

山本幸生¹、三浦昭¹、平田成²、石川博³

¹ 宇宙航空研究開発機構 宇宙科学研究所

² 会津大学 宇宙情報科学研究センター

³ 東京都立大学

※ 本研究は2019年度 会津大学宇宙情報科学研究センター共同研究との研究です。

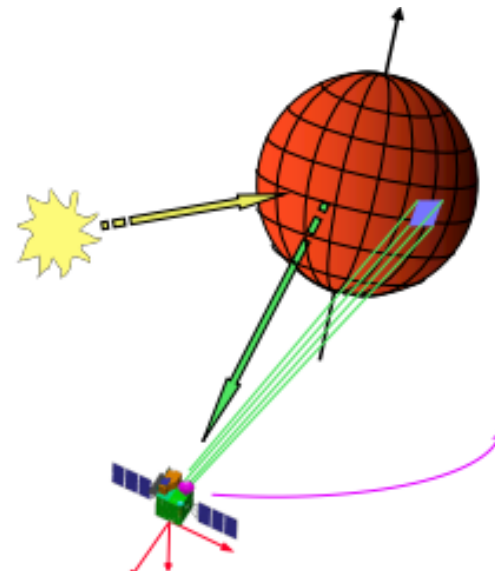
SPICEとは?

NASA JPLが開発した惑星探査で必要となる幾何学計算を行うためのライブラリ群/アプリケーション/データ(カーネル)を提供

宇宙機の運用・解析に必要な様々な計算が可能

- ・ 惑星や月などの高精度な位置/速度
- ・ 惑星や月などのサイズ/形状/姿勢
- ・ 宇宙機の姿勢や動き
- ・ 観測機器の視野角

etc.

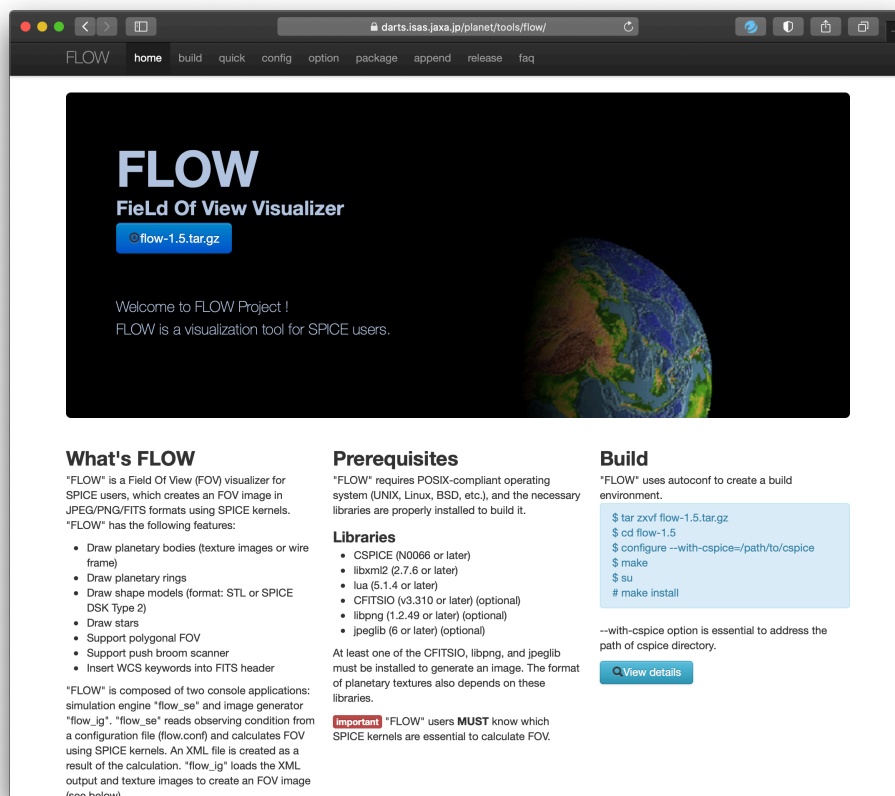


(C)NASA

<https://naif.jpl.nasa.gov/naif/>

2011年ごろ～

- NASA JPLが開発したSPICE Toolkitを用いた視野の可視化ツールを開発
- Field Of View Visualizer: FLOW
- 特徴
 - SPICEでデータ公開されれば可視化可能
 - 衛星運用に耐えられる品質
 - 形状モデルの描画
 - 視野角形状が多角形や丸に対応
 - 幾何学計算はSPICEをベースに
 - WMSによる地形画像配信に対応
- 開発
 - C言語
 - 描画はOpenGLライクな独自エンジン



<https://darts.isas.jaxa.jp/planet/tools/flow/>

OpenGLを使わなかったことで . . .

- メリット

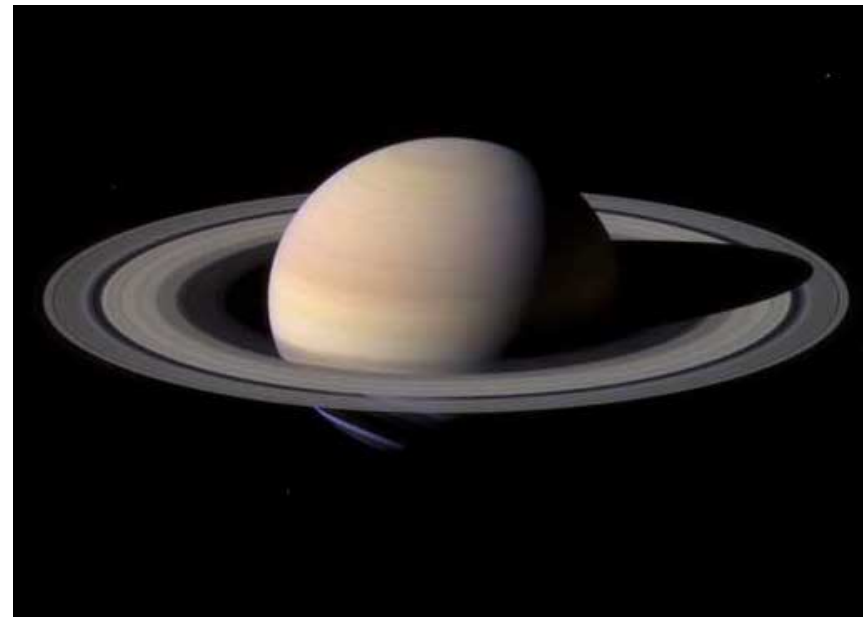
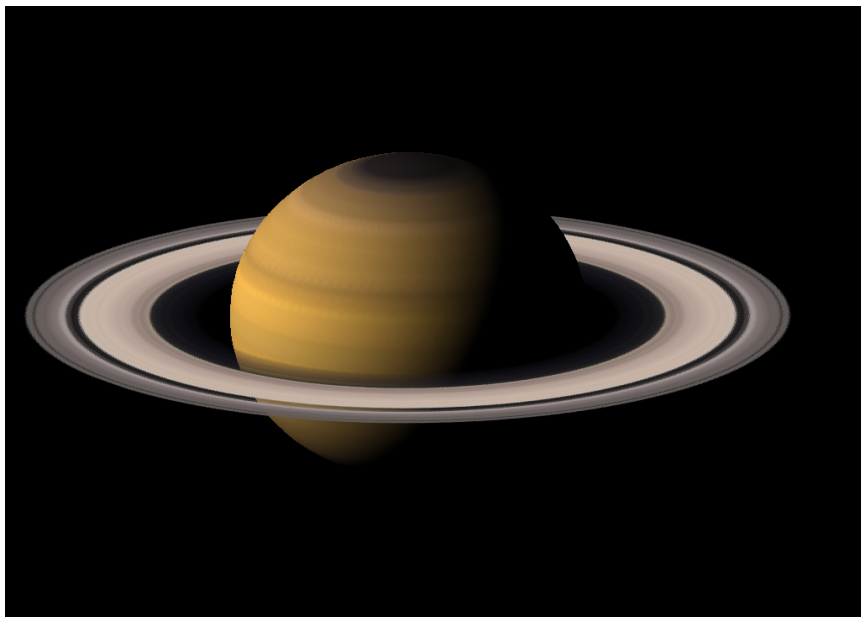
- 当時流行の「Hadoop」を使って数1000枚の画像描画を分散化
- マルチプラットフォーム化が容易(Windows, MacOS, Linux)
- ウェブサービスのバックエンドとして利用可能
- 描画ライブラリは独自のため長期安定

- デメリット

- CPUしか使っていないため遅い
- 凝ったレンダリングが困難(凝ったレイトレーシング等はなし)

Cassiniが2004年3月27日に撮像した土星

左：FLOWによるシミュレーション画像 右：実画像



Hayabusa2の運用ツールとしても活用

Flow on the Web：日付を入力してDrawボタンを押すだけ

FLOW on the web Hayabusa2 ONC Planning Hayabusa2 TIR Planning Hayabusa2 ONC Hayabusa2 TIR Links

Hayabusa2 ONC Simulator

Hayabusa2 ONC Simulator makes a simulated image on the web using FLOW. The usage of the simulation is [here](#).

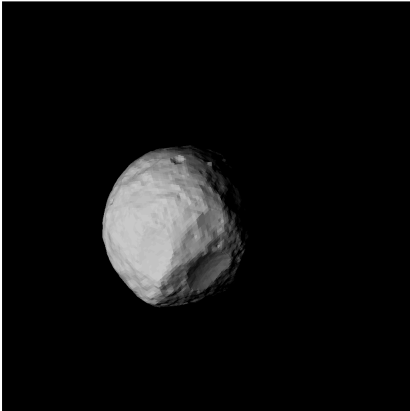
Date-time:

Camera: ONC-T ONC-W1 ONC-W2

Attitude:
 HK/AOCSC
 +Zsc to the Sun +Zsc to the Earth
 -Zsc to the Earth -Zsc to the Moon -Zsc to Ryugu
 Quaternion (J2000.space-delimiter)

Shape model:
v20190308:
 SPC 49k SFM 200k SFM 800k SFM 3M
v20190130:
 SFM 47k SFM 187k SFM 750k SFM 6M
v20181204:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20181109:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20181014:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20180829:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20180816:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20180810:
 SPC 49k SPC 200k SPC 800k SPC 3M
.....

2019-06-26T03:46:00 (normal)



rotate: 0 deg

Ryugoid (Ryugu到着前の模擬小惑星)によるシミュレーション

FLOW on the web Hayabusa2 ONC Planning Hayabusa2 TIR Planning Hayabusa2 ONC Hayabusa2 TIR Links

Hayabusa2 ONC Simulator

Hayabusa2 ONC Simulator makes a simulated image on the web using FLOW. The usage of the simulation is [here](#).

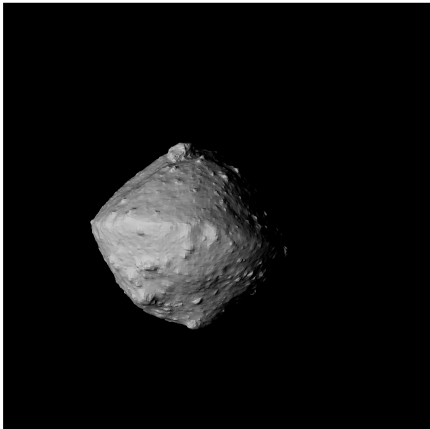
Date-time:

Camera: ONC-T ONC-W1 ONC-W2

Attitude:
 HK/AOCSC
 +Zsc to the Sun +Zsc to the Earth
 -Zsc to the Earth -Zsc to the Moon -Zsc to Ryugu
 Quaternion (J2000.space-delimiter)

Shape model:
v20190308:
 SPC 49k SFM 200k SFM 800k SFM 3M
v20190130:
 SFM 47k SFM 187k SFM 750k SFM 6M
v20181204:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20181109:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20181014:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20180829:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20180816:
 SPC 49k SPC 200k SPC 800k SPC 3M
v20180810:
 SPC 49k SPC 200k SPC 800k SPC 3M
.....

2019-06-26T03:46:00 (normal)



Ryugu形状モデルによるシミュレーション

2011年から開発した視野シミュレータFLOWは一定の成功をおさめた

周辺状況の変化

- プログラミング言語
 - 機械学習によりPythonが浸透
- SPICE Toolkit for Python
 - 有志作成のPython版SPICEが登場(SpiceyPy since 2014~)
- 解析環境
 - Jupyter Notebookによるクイックな解析環境
- GPUの無償提供
 - Google Colab上で制限つきながらも利用可能に

SPICEデータ (Kernel)の提供方式の違い

- 2011年当時

- 個々のファイルを別々にダウンロード
- 必要なカーネルはユーザが適切にダウンロードして利用

- 2021年現在

- SPICEカーネルの提供者による(ベストなセットの)メタカーネルの提供
- カーネルは全てダウンロードしメタカーネルのパス指定のみ変えて利用

目的

- 近年の状況に合わせたPython版の視野シミュレータを開発する
 - 目的に適したグラフィックライブラリを選定する
 - なるべくPythonらしく使えるように設計したい

開発要求の違い

- 2011年当時

- マルチプラットフォームで
- C言語をベースに
- OpenGLは使わずに
- Hadoopで分散処理可能で
- Webサービスで簡単に

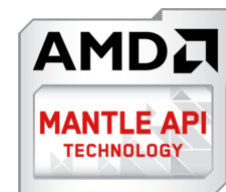
- 2021年現在

- マルチプラットフォームで
- Pythonをベースに
- GPUを使って高速に
- Jupyter Notebookで簡単に

現在のニーズに合わせて
2019年度、会津大学との共同研究により再開発

グラフィックエンジンの標準化を取り巻く状況

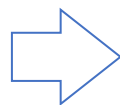
- 2008年 OpenGL 3.0で上位互換性を放棄
 - 馴染み深いglBegin()/glEnd()の時代は終焉しシェーダの時代へ
- 2014年 ローレベルAPIの標準3D API規格 Vulkanの発表
- 2019年11月 NVIDIAのCUDA ToolkitはmacOS非サポート
 - Mojave以降、NVIDIAのドライバがインストールできない
 - AppleがOpenGL/OpenCLからMetal APIへ移行推奨
 - ※ CUDAはGPUを画像だけでなく汎用的に使うGPGPUのためのもの
- 2021年「Vulkan」の対応状況
 - OSメーカーは独自路線、グラフィックボードメーカーが対応ドライバ提供
 - Windows「DirectX」、Apple「Metal」を推奨
 - AMD「Mantle」をベースに「Vulkan」に対応
 - NVIDIA「Vulkan対応のドライバ」を「Windows」と「Linux」に提供
 - 標準化団体のKhronosがmacOSに「Vulkan」を提供



技術選定の葛藤

所感

- ・ ローレベルなAPIと付き合うのは大変そう
- ・ OpenGLはなんだかんだで息が長そう
- ・ Vulkanが浸透するにはまだまだ時間が必要そう



初心に返って元々の要求は?

1. **マルチプラットフォームで動作すること**
2. **Pythonを用いること**
3. **GPUを用いて描画できること**

Python パッケージ	マルチプラットフォーム	Jupyter Notebook	Google Colaboratory	CPU/GPU 描画できること
ipycanvas	○	○	×	×
opencv2	○	○	△ (CPUのみ)	○
トレードオフから見るとPythonではOpenGLが良さそうだが・・・ PyOpenGL + α	○	○	○	○

各種ハードウェアにおけるOpenGLバージョン

プラットフォーム	GPU	GL_VERSION	GL_VENDOR	GL_SHADING_LANGUAGE_VERSION
Google Colaboratory (2020.03.20現在)	Tesla T4	4.6.0 NVIDIA 418.67	NVIDIA Corporation	4.60 NVIDIA
iMac (Retina 5K, 27-inch, 2017)	Radeon Pro 570 4GB	2.1 ATI-3.5.5	ATI Technologies Inc.	1.20
Windows 10	RTX-2070 SUPER	4.6.0 NVIDIA 442.59	NVIDIA Corporation	4.60 NVIDIA
iMac (27-inch, 2009)	ATI Radeon HD 4850 512MB	2.1 ATI-10.4.14	ATI Technologies Inc.	1.2

iMacは宇宙関連の分野で主要なマシンの一つであるため、ここから利用可能なOpenGLのバージョンは2.1、
OpenGLのバージョンは1.2ということになる。

結局どうしたか？

- 低いレイヤーのAPIは使わず pyrender というパッケージを使用
 - 形状モデルのOBJやSTLの描画に対応
 - ヘッドレスのレンダリングに対応 (OSMesa / EGL)

この選択が正しかったかどうかは今だにわからない・・・

Python版FLOWデモ on Google Colab.

- (1)Googleドライブのマウント
- (2)SPICEカーネルを読み込み
- (3)シミュレーション
- (4)シミュレーション結果にテクスチャの割当
- (5)描画

(1) Google Driveのマウント



Kaguya HDTV Rising Earth ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 最終編集: 2020年4月25日

+ コード + テキスト

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
▶ import sys
sys.path.append('/content/drive/My Drive/flow/python')
```

```
▶ !pip install -q spiceypy pyrender
```

```
[ ] import os
os.environ['PYOPENGL_PLATFORM'] = 'egl'
```

```
[ ] import spiceypy as spice
import spiceflow as flow
```

(2)SPICEカーネルを読み込み

インターネット上に置かれたメタカーネルを元にダウンロード & 読み込みを行うremote_furnsh()

```
url = "http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/extras/mk/SEL_V02.TM"  
flow.remote_furnsh(url, "selene_colab.tm", local_dir="/content/drive/My Drive/flow/kernels/SELENE")
```

```
↳ http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/lsk/naif0009.tls ==> /content/drive/My Drive/flow/kernels/SELENE/data/lsk/naif0009.tls  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/pck/pck00010.tpc ==> /content/drive/My Drive/flow/kernels/SELENE/data/pck/pck00010.tpc  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/pck/moon_pa_de421_1900-2050.bpc ==> /content/drive/My Drive/flow/kernels/SELENE/data/pck/moon_pa_de421_19  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/sclk/SEL_M_V01.TSC ==> /content/drive/My Drive/flow/kernels/SELENE/data/sclk/SEL_M_V01.TSC  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/fk/moon_080317.tf ==> /content/drive/My Drive/flow/kernels/SELENE/data/fk/moon_080317.tf  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/fk/moon_assoc_me.tf ==> /content/drive/My Drive/flow/kernels/SELENE/data/fk/moon_assoc_me.tf  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/fk/SEL_V01.TF ==> /content/drive/My Drive/flow/kernels/SELENE/data/fk/SEL_V01.TF  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/ik/SEL_HDTV_V01.TI ==> /content/drive/My Drive/flow/kernels/SELENE/data/ik/SEL_HDTV_V01.TI  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/ik/SEL_LALT_V01.TI ==> /content/drive/My Drive/flow/kernels/SELENE/data/ik/SEL_LALT_V01.TI  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/ik/SEL_MI_V01.TI ==> /content/drive/My Drive/flow/kernels/SELENE/data/ik/SEL_MI_V01.TI  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/ik/SEL_SP_V01.TI ==> /content/drive/My Drive/flow/kernels/SELENE/data/ik/SEL_SP_V01.TI  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/ik/SEL_TC_V01.TI ==> /content/drive/My Drive/flow/kernels/SELENE/data/ik/SEL_TC_V01.TI  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/ik/SEL_XRS_V01.TI ==> /content/drive/My Drive/flow/kernels/SELENE/data/ik/SEL_XRS_V01.TI  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/ck/SEL_M_ALL_S_V03.BC ==> /content/drive/My Drive/flow/kernels/SELENE/data/ck/SEL_M_ALL_S_V03.BC  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/ck/SEL_M_ALL_D_V03.BC ==> /content/drive/My Drive/flow/kernels/SELENE/data/ck/SEL_M_ALL_D_V03.BC  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/spk/de421.bsp ==> /content/drive/My Drive/flow/kernels/SELENE/data/spk/de421.bsp  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/spk/SEL_M_071020_090610_SGMH_02.BSP ==> /content/drive/My Drive/flow/kernels/SELENE/data/spk/SEL_M_071020  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/spk/SEL_M_071020_081226_SGMI_05.BSP ==> /content/drive/My Drive/flow/kernels/SELENE/data/spk/SEL_M_071020  
http://darts.isas.jaxa.jp/pub/pds3/sln-l-spice-6-v1.0/slntp_1000/data/spk/SEL_MAIN_GRGM900C_L660_DIRALT_20081030-20090610.bsp ==> /content/drive/My Drive/flow/kernels/SELENE/
```

(3) シミュレーション

```
[7] et = spice.utc2et("2008-04-05T21:44:26")
```

```
[8] obsinfo = flow.simulate(  
    inst="SELENE_HDTV_TELE",  
    et=et,  
    abcorr="LT+S",  
    obsrvr="SELENE",  
    width=1920,  
    height=1080,  
    mag_limit=4.0,  
)
```

obsinfoの中身はXML

```
import xml.etree.ElementTree as ET  
import xml.dom.minidom as minidom  
xml = obsinfo.to_xml()  
print(minidom.parseString(ET.tostring(xml)).toprettyxml(indent="  "))
```

```
<?xml version="1.0" ?>  
<svdoc>  
  <frame>  
    <view>  
      <date>2008-04-05T21:44:26.000</date>  
      <location>  
        <x>8921217.568806</x>  
        <y>70794665.081172</y>  
        <z>131199026.976252</z>  
      </location>  
      <boresight>  
        <x>0.000000</x>  
        <y>0.000000</y>  
        <z>1.000000</z>  
      </boresight>  
      <bounds>  
        <oints>
```

(4) テクスチャの割当

```
obsinfo.set_obs_table(  
  {  
    "PLANET.EARTH": {  
      "type": "texture-body",  
      "file": "/content/drive/My Drive/flow/data/texture/earth.jpg",  
    },  
    "MOON": {  
      "type": "texture-body",  
      "file": "/content/drive/My Drive/flow/data/texture/2k_moon.jpg",  
    }  
  }  
)
```

マイドライブ > flow > data ▾

名前

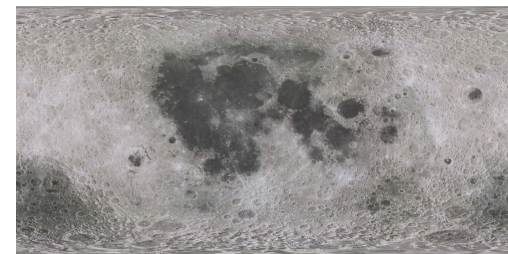
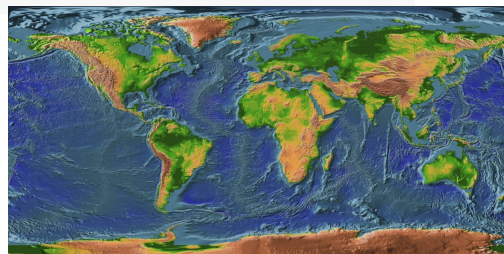
models

kernels

texture

earth.jpg

moon.jpg



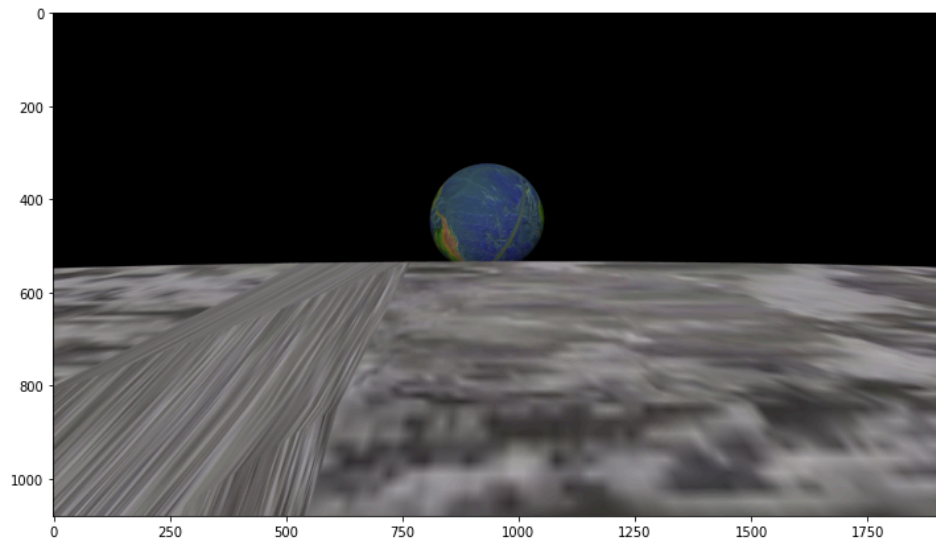
(5) 描画

```
[23] import matplotlib.pyplot as plt
```

```
[24] img = flow.render(obsinfo, bg_color=[0.0, 0.0, 0.0], wireframe=False)
```

```
[25] plt.figure(figsize=(14, 7))  
plt.imshow(img)
```

<matplotlib.image.AxesImage at 0x7fe54f5480b8>



まとめ

- GPU描画に関する周辺状況を調査しOpenGLを選択し、最終的にはpyrenderを用いてEGLによる描画とした
- Google Colab.上で動作するPython版を開発した
 - `remote_furnsh()`によるインターネット上のカーネル自動取得
 - Pythonらしい記述でレンダリング