

大気補正プログラムのコンテナ化とその利用

飯倉善和¹、久世宏明²、関口美保³、眞子直弘⁴



補正前 (DN)



補正後 (反射率)

LANDSAT7(ETM+)
2002.6.30p107r32

- 1 弘前大学大学院 理工学研究所
- 2 千葉大学環境 リモートセンシング研究センター
- 3 東京海洋大学大学院 海洋科学技術研究所
- 4 JAXA 宇宙科学研究所

研究の背景 (光学衛星画像の解析処理)

衛星画像 => 幾何補正 => 相対補正 (土地被覆分類・変化抽出)
=> 絶対補正 (反射率とエアロゾルの同時推定)
画像処理、GIS、放射伝達モデル、統計モデル (機械学習)
学際的かつ組織的な取り組みの必要性
=> 情報通信技術 (ICT) の利用

衛星画像処理プログラムのオープンソース化

プログラミング言語: Python (FORTRAN, C++ => Swift, Go, Scala?)
分散型バージョン管理システム: Git
共有リポジトリでの公開: GitHub
=> ライブラリ (パッケージ) などの利用

計算機環境の整備

コンテナ化: Docker & DockerHub
クラウドの利用: Google Cloud Platform

飯倉: 衛星画像処理プログラムのオープンソース化について —GitHubを利用した公開—, 日本リモートセンシング学会誌, 36(4), 2016

1. はじめに

ロボット (ROS)、医用画像 (dcmk/Osirik)、地理情報 (gdal/QGIS)
画像処理 (OpenCV)、衛星画像処理 (???)

2. オープンソース化の背景

利用者: 巨人の肩に乗る、車輪の再発明は必要ない
開発者: 集合知の利用、技術のリードと標準化

3. 利用と公開に必要な知識

3.1 プログラミング言語: Python

動的なスクリプト言語
ライブラリが充実

3.2 バージョン管理システム:

集中型 => 分散型: Git

3.3 GitHub利用の注意点

READMEの作成
ライセンスの設定

(コピーレフト => パーミッシブル)

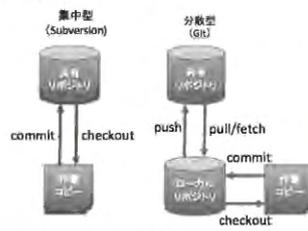


図1 バージョン管理システムの構造と操作

公開したプログラムの概要

<https://github.com/y-iikura>

AtmosphericCorrection: 幾何補正済みの衛星画像から地表反射率とエアロゾルの光学的厚さの空間分布の複数の可能性を同時に推定する。放射伝達コードとして6Sを利用。

KibanDEM: 基盤地図情報・数値標高モデルから指定した範囲をラスタ形式で切り出すプログラム。範囲の指定は緯度経度およびUTM座標 (ゾーン54) で行える。

JaxaLandcover: Jaxaの高解像度土地利用土地被覆図 (約10m解像度) から指定した範囲をラスタ形式で切り出すプログラム。範囲の指定は緯度経度およびUTM座標 (ゾーン54) で行える。

LandsatETM: ランドサット7号ETM+(Level 1T) から指定した範囲を切り出す。位相限定相関法を用いて、数値標高モデルから作成した陰影画像との位置ズレの評価も行える。

LandsatOLI, AlosAvnir2 など

開発環境: MacbookPro (Mac OS 10.12.6)
CPU (2.4GHz Intel Core i5) メモリ (4GB)
Python 2.7, gfortran, GDAL, OpenCV
(Macports => Homebrew)

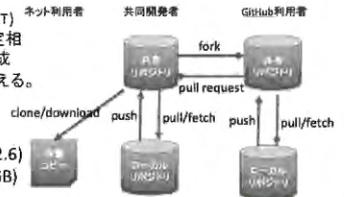


図3 GitHubを用いた共同開発

衛星画像処理プログラムの開発と利用のための計算機環境

基本となる計算機環境 (オープンソース)

ホストとしてどんな計算機でも使える (Mac, Windows, Linux)

仮想マシン (コンテナ) として Linux を利用する。

ディストリビューションの違いに注意

プログラミング言語として Python 2.7, GCC, GFortran を利用

ライブラリ (パッケージ) として

GS & RSTAR (放射伝達コード), GDAL, OpenCV

Python パッケージ (numpy, pyproj, scipy, scikit-learn, ...)

パッケージ管理システム

Debian系 (Ubuntu): dpkg => apt

RedHat系 (Centos): rpm => yum

conda => Anaconda (Python パッケージのディストリビューション)

計算機環境のコンテナ化

Pythonリポジトリ PyPI にアクセス

chroot (UNIXファイルシステムの隔離 1979) => jail (プロセス空間の隔離 1998)

クラウドの利用 (Infrastructure as a Service)

Amazon: EC2 (AWS)

Google: Google Computer Engine (GCP)

Microsoft: Virtual Machine (Azure)

仮想マシンとコンテナ

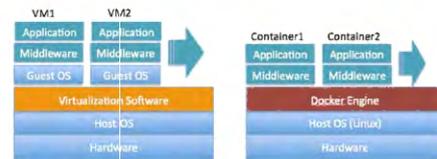


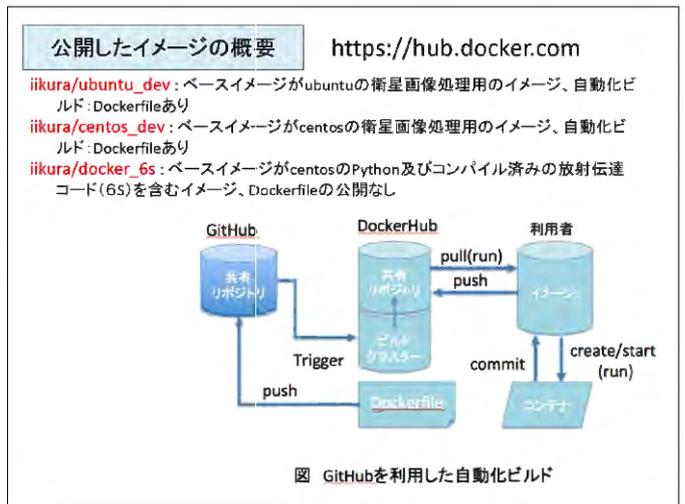
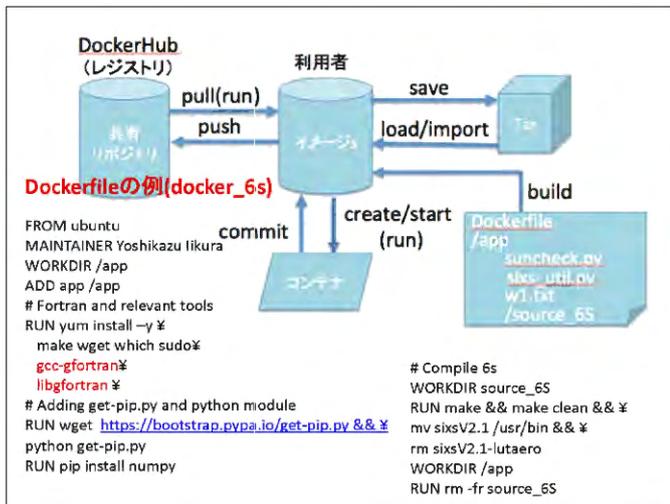
Fig.1 Comparison of Virtual Machine and Container

MacとWindowsでの実現



Fig.3 Docker for Mac

Fig.4 Docker for Windows



コンテナの起動と利用例

docker_6s: コンパイル済みの6S(sixsv2.1)と入力テンプレート(w1.txt)、6sを利用するためのPythonプログラム(suncheck_6s.py)とモジュール(sixs_util.py)。

```

$ docker run --rm -it iikura/docker_6s
[root@... app]# which sixsv2.1
/usr/bin/sixsv2.1
[root@... app]# cat w1.txt | sixsv2.1

```

```

***** 6SV version 2.1 *****
*
*          geometrical conditions identity
*          -----
*          user defined conditions
*
*
* month:   day :
* solar zenith angle:  26.20 deg  solar azimuthal angle:  123.60 deg
* view zenith angle:   3.00 deg   view azimuthal angle:   281.49 deg
* scattering angle:   153.00 deg  azimuthal angle difference: 157.89 deg
*
[root@... app]# python suncheck.py

```

```

[2080.2292508953365, 2010.0367069475109, 1981.4157117648372, 1946.8382139426833,
1919.1404766691237, 1003.6833523136854, 1904.99370290037, 1922.8335063055061,
1953.0478679443364, 1986.6574340875734, 2016.499512311346, 2032.3449093323505]

```

まとめと今後の課題

衛星画像処理プログラムのオープンソース化
 プログラミング言語: Python (FORTRAN, C++ => Swift, Go, Scala?)
 分散型バージョン管理システム: Git
 共有リポジトリでの公開: GitHub
 => ライブラリ(パッケージ)などの利用

計算機環境の整備
 コンテナ化: Docker & DockerHub
 ubuntu_dev, centos_dev, conda_dev, docker_6s, docker_rstar
 クラウド&GPUの利用: Google Cloud Platform

地表面反射率とエアロゾル光学的厚さの同時推定法
 解の不定性の解消(地上観測、整合性の評価)
 照り返し光・天空光・方向性反射などの取り扱い
 MODISプロダクトの改良への利用
 エアロゾルタイプ及び高度分布の推定
 物理モデルと統計モデル(機械学習)の融合