

Image Data Cube

2019-Oct-08

Rennan Marujo



50 km
20 mi



2012

Fast

1984

Google

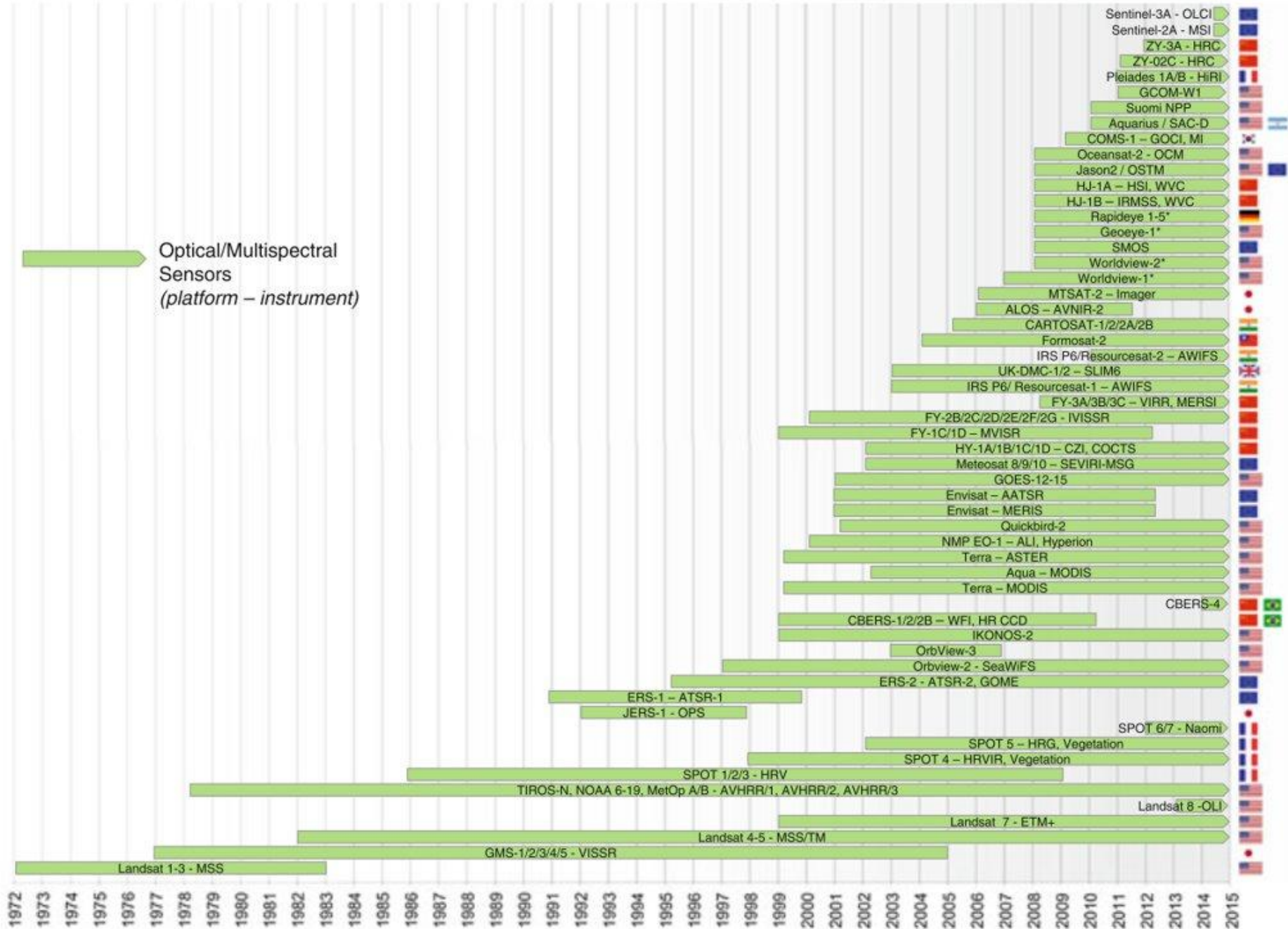


2012



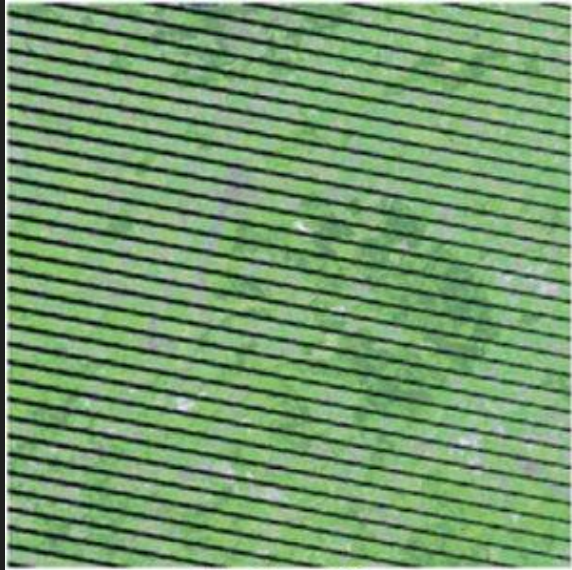
Spatial Ground Resolution	Ultra High < 1m	Very High 1 – 4 m	High 4 – 10 m	Medium 10 – 50 m	Low 50 – 250 m	Very Low > 250 m
---------------------------	--------------------	----------------------	------------------	---------------------	-------------------	---------------------



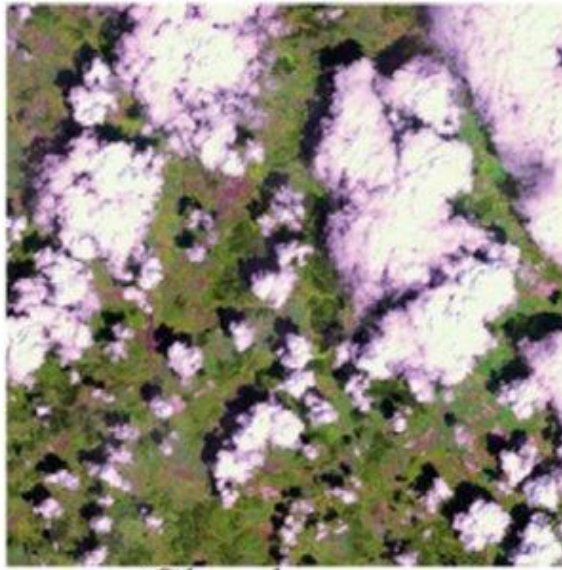


Four small satellite icons are positioned in the corners of the slide. Top-left: A satellite with a yellow body and black solar panels. Top-right: A satellite with a yellow body and blue solar panels. Bottom-left: A satellite with a red and orange body and blue solar panels. Bottom-right: A satellite with a yellow body and blue solar panels.

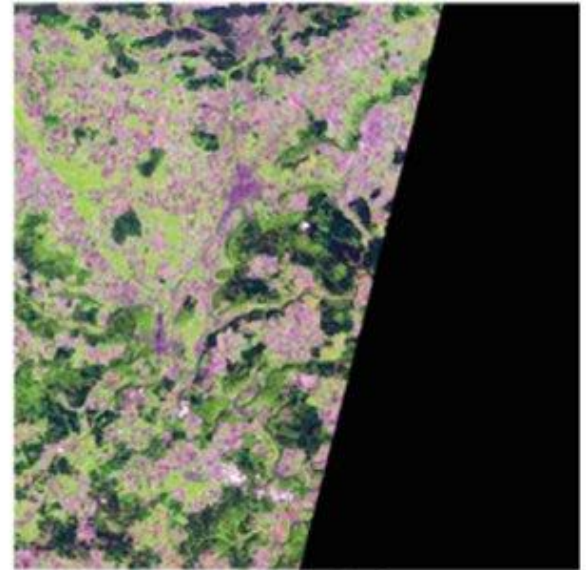
DATA MUST BE
COMPATIBLE



SLC off



Cloud cover



Partial



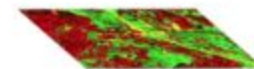
Pre-processing

Registration
Radiometric Correction
Cloud Detection



Processing

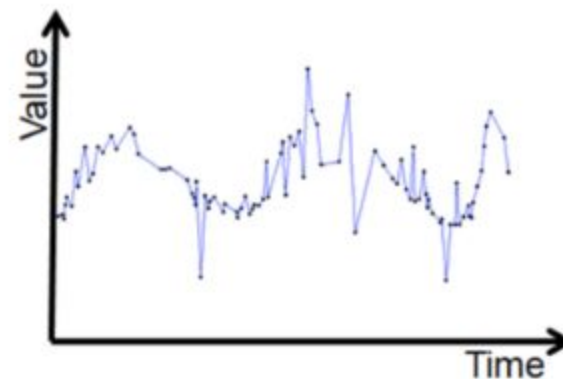
Pixel
Segmentation
Time series

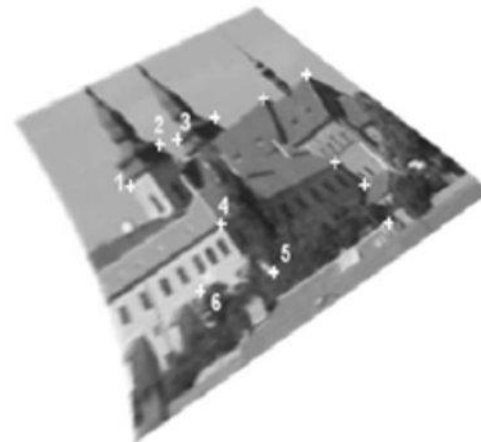
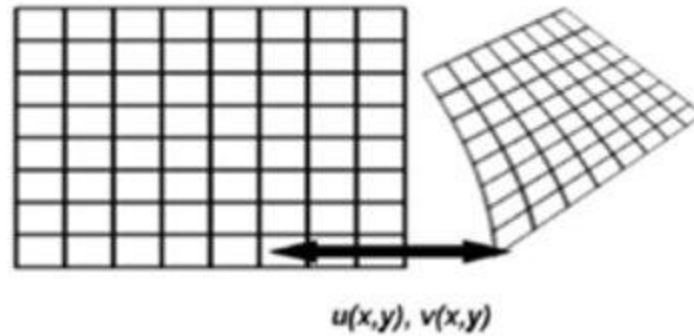


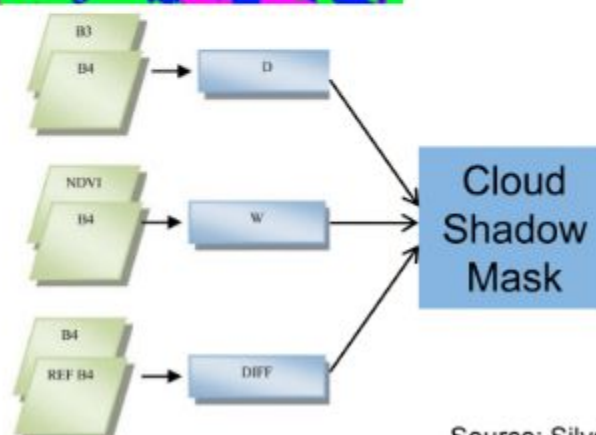
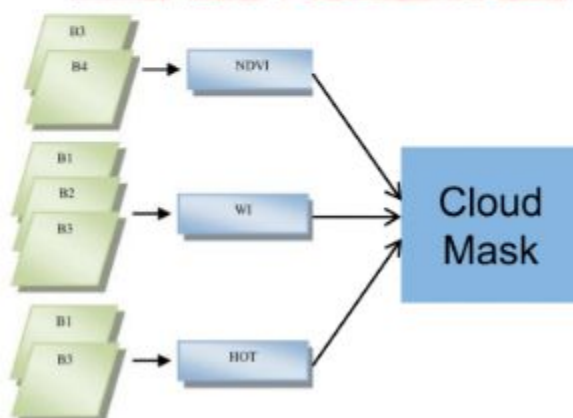
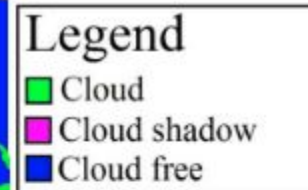
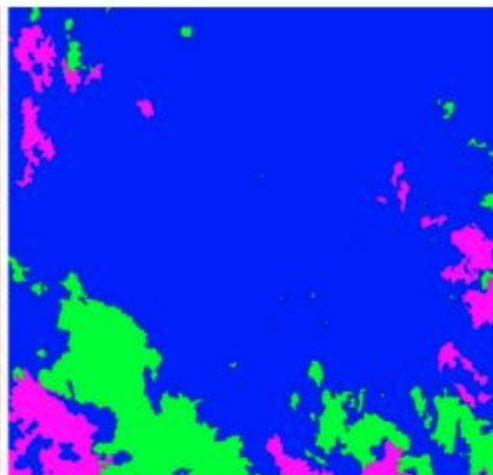
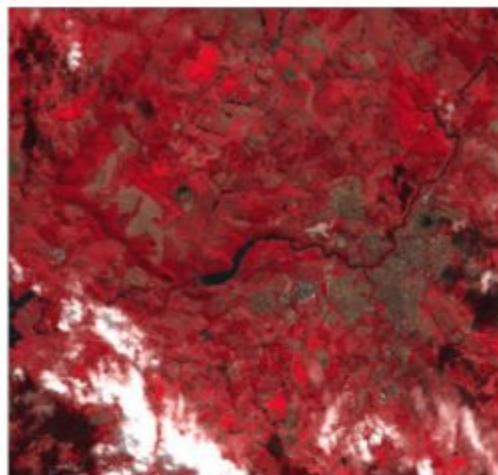
Classification

Training
Classification
Validation

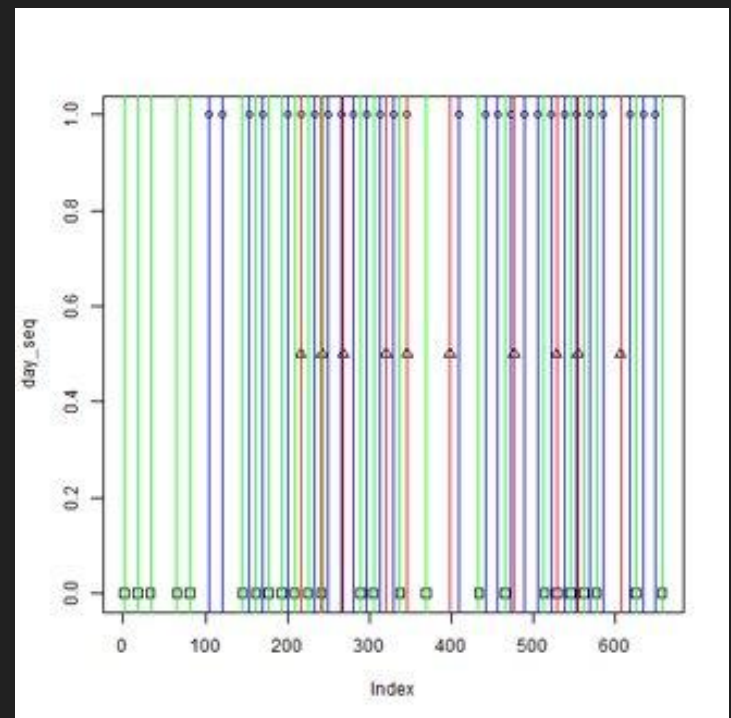
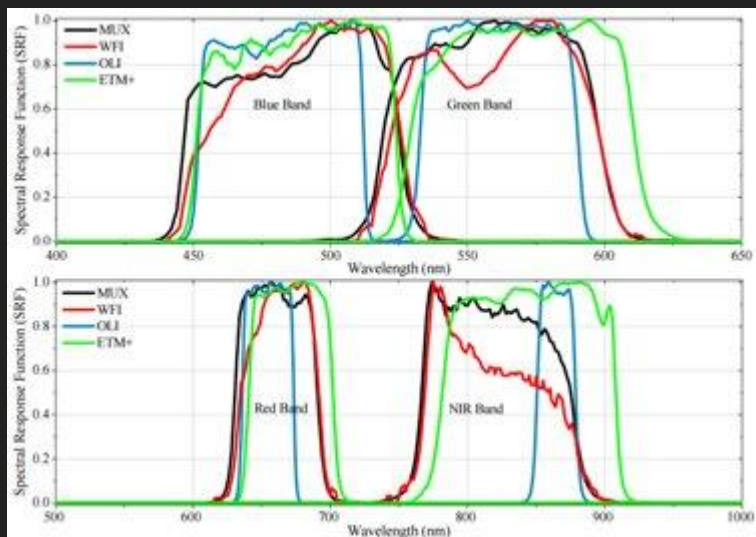
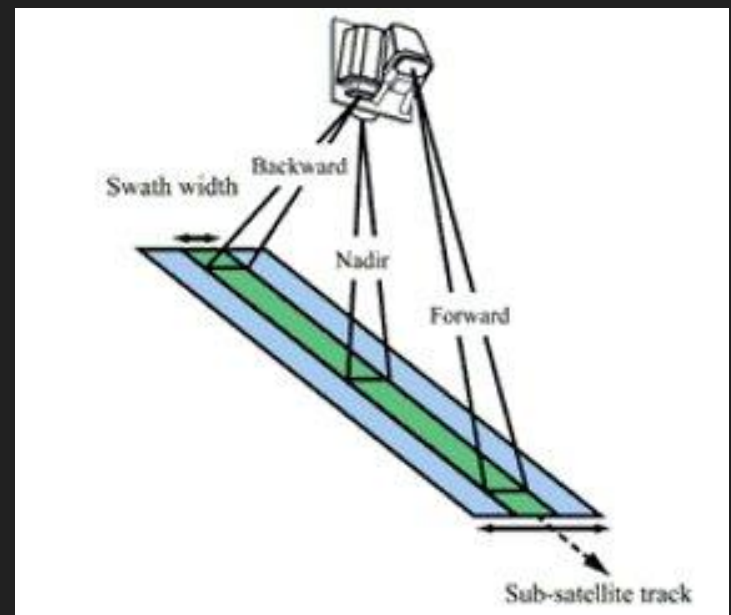
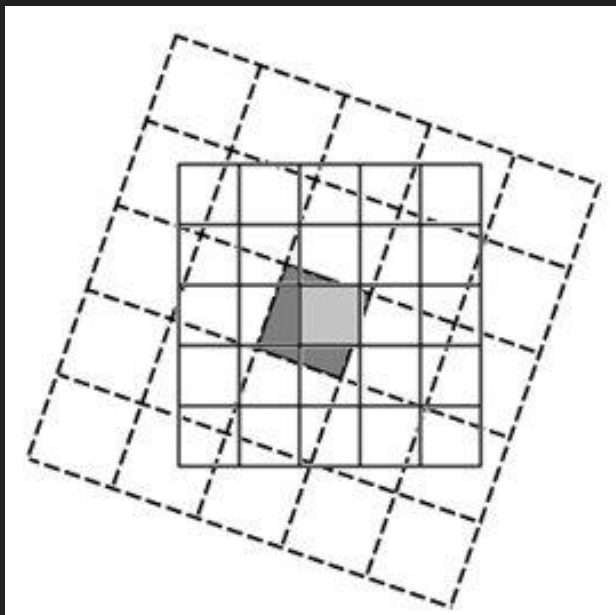
Multi-source

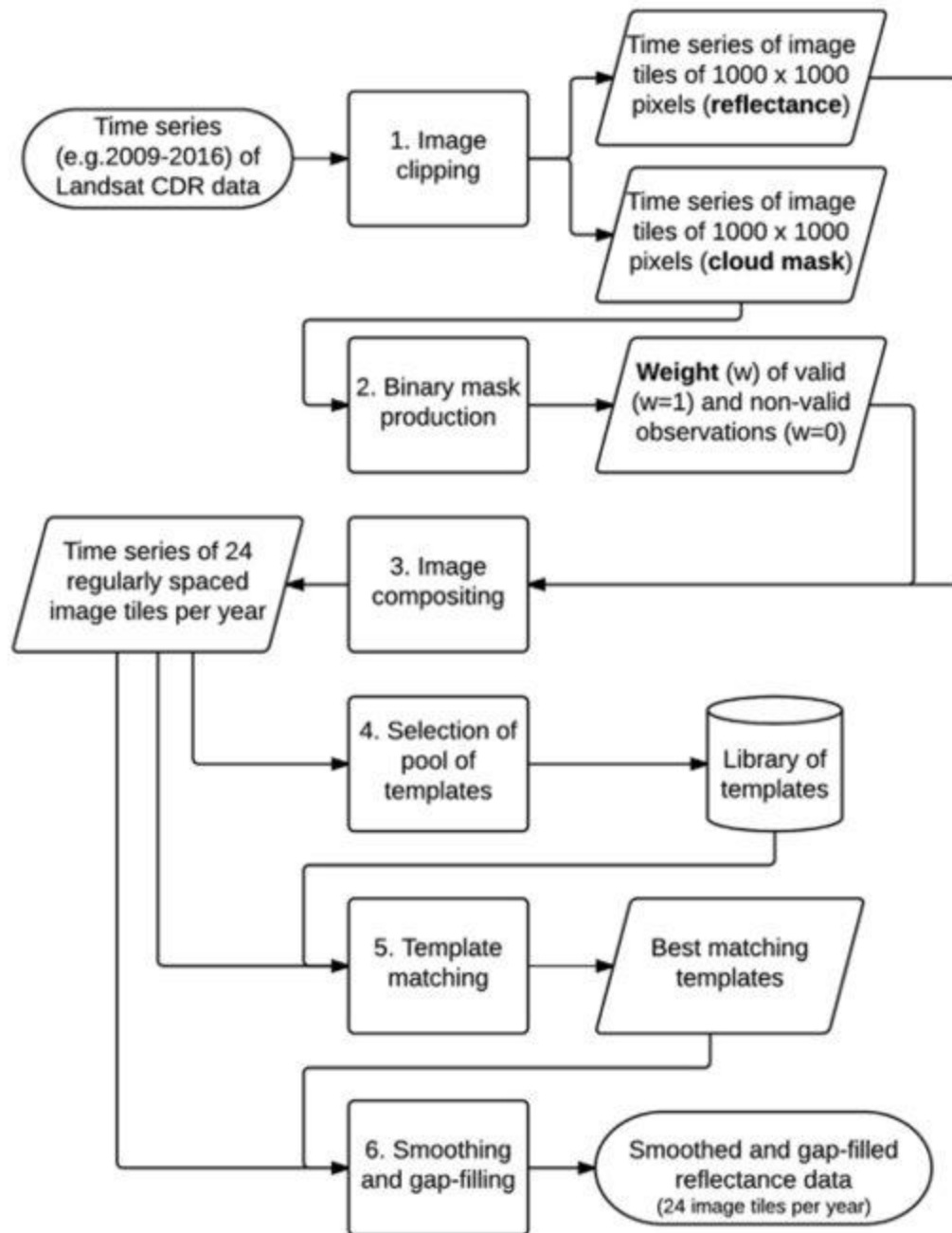


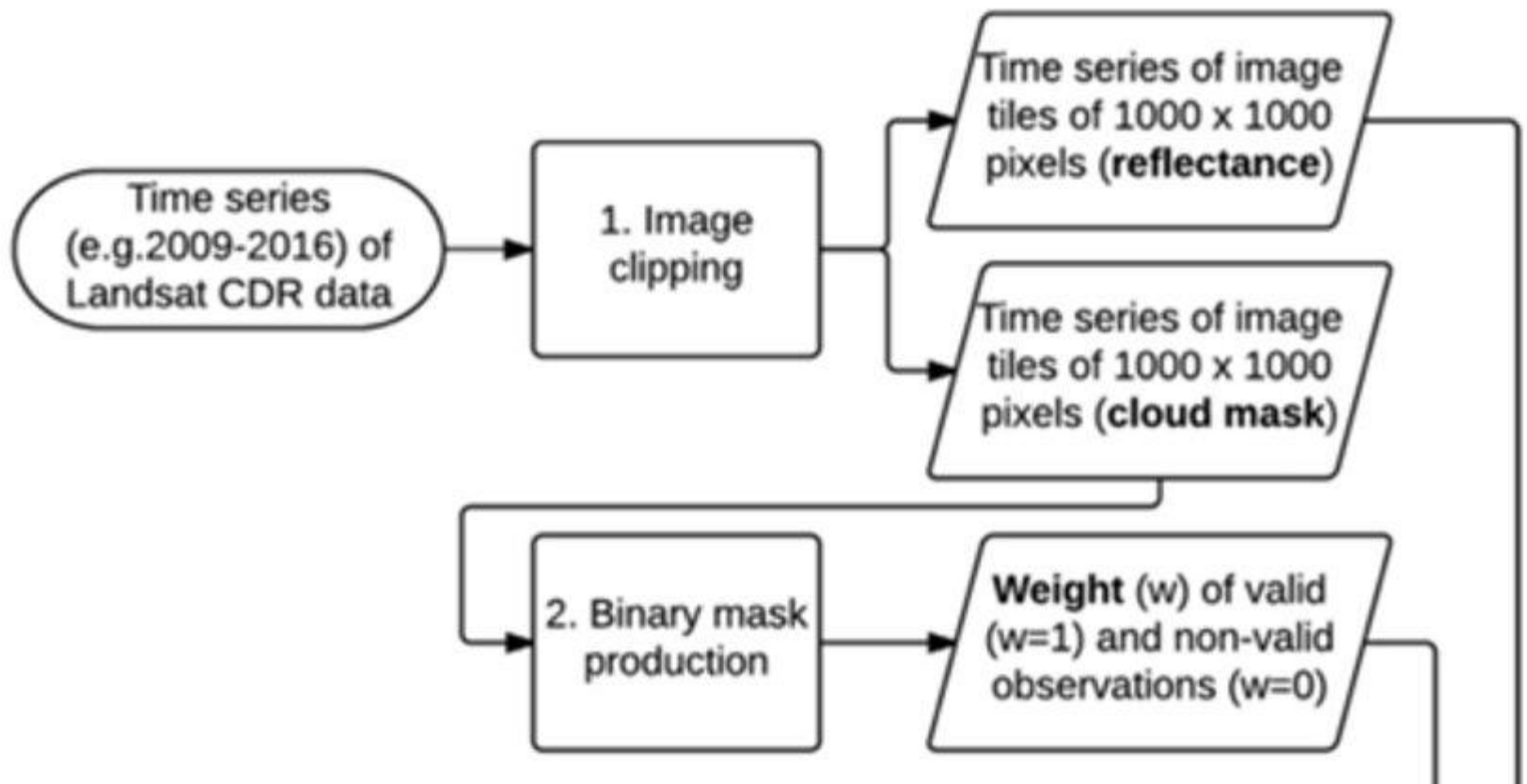


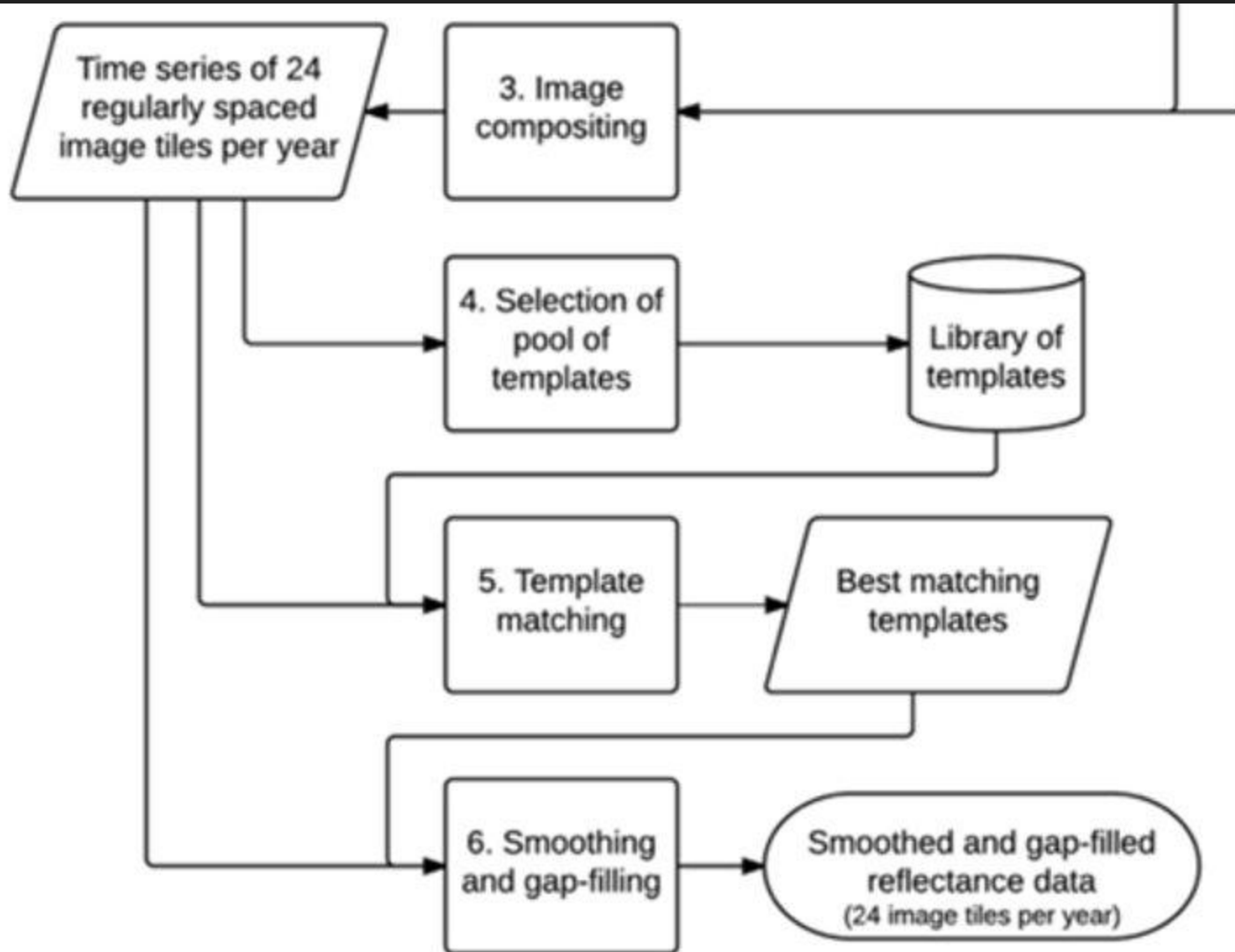


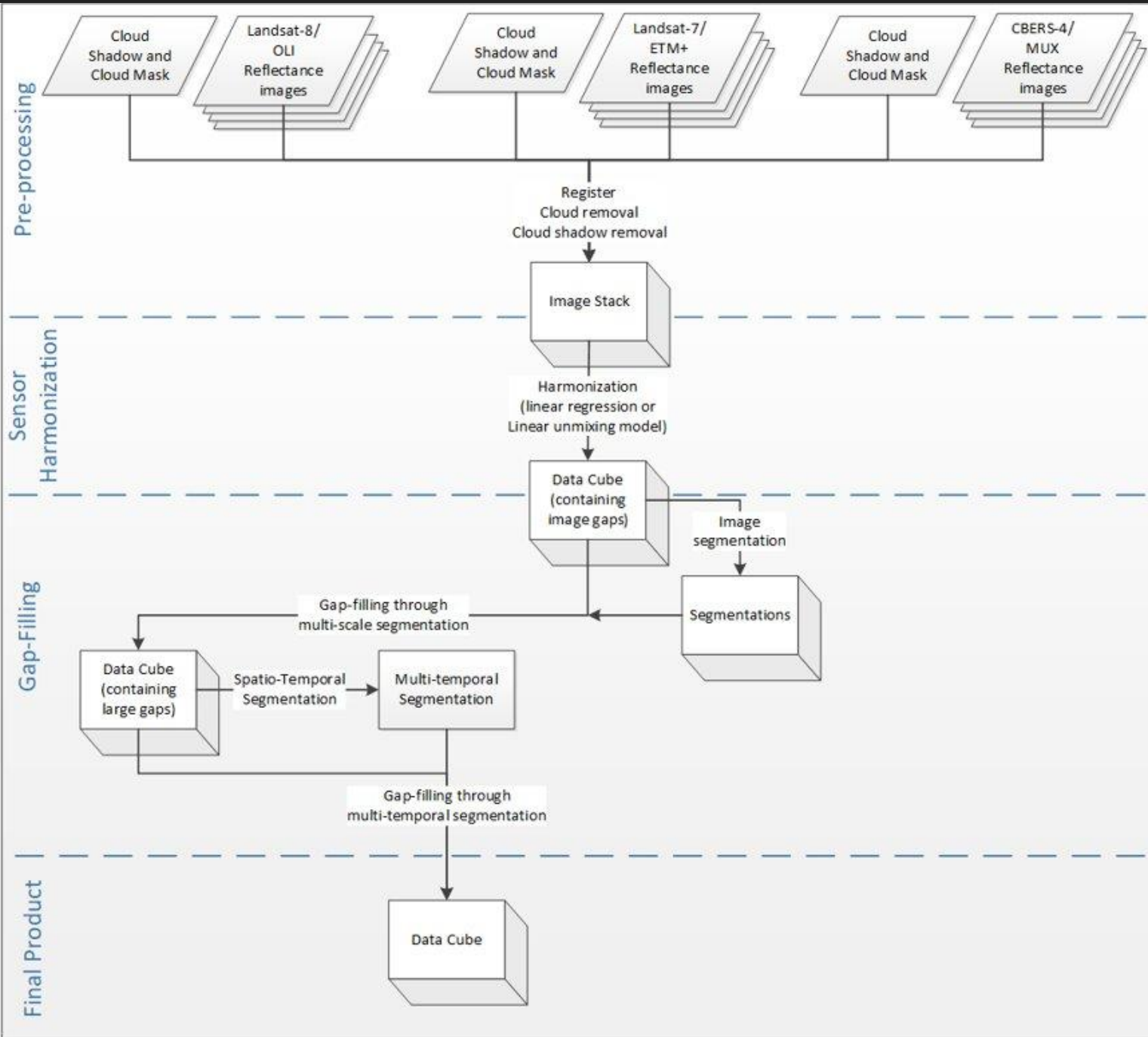
Source: Silva et. Al (2016)

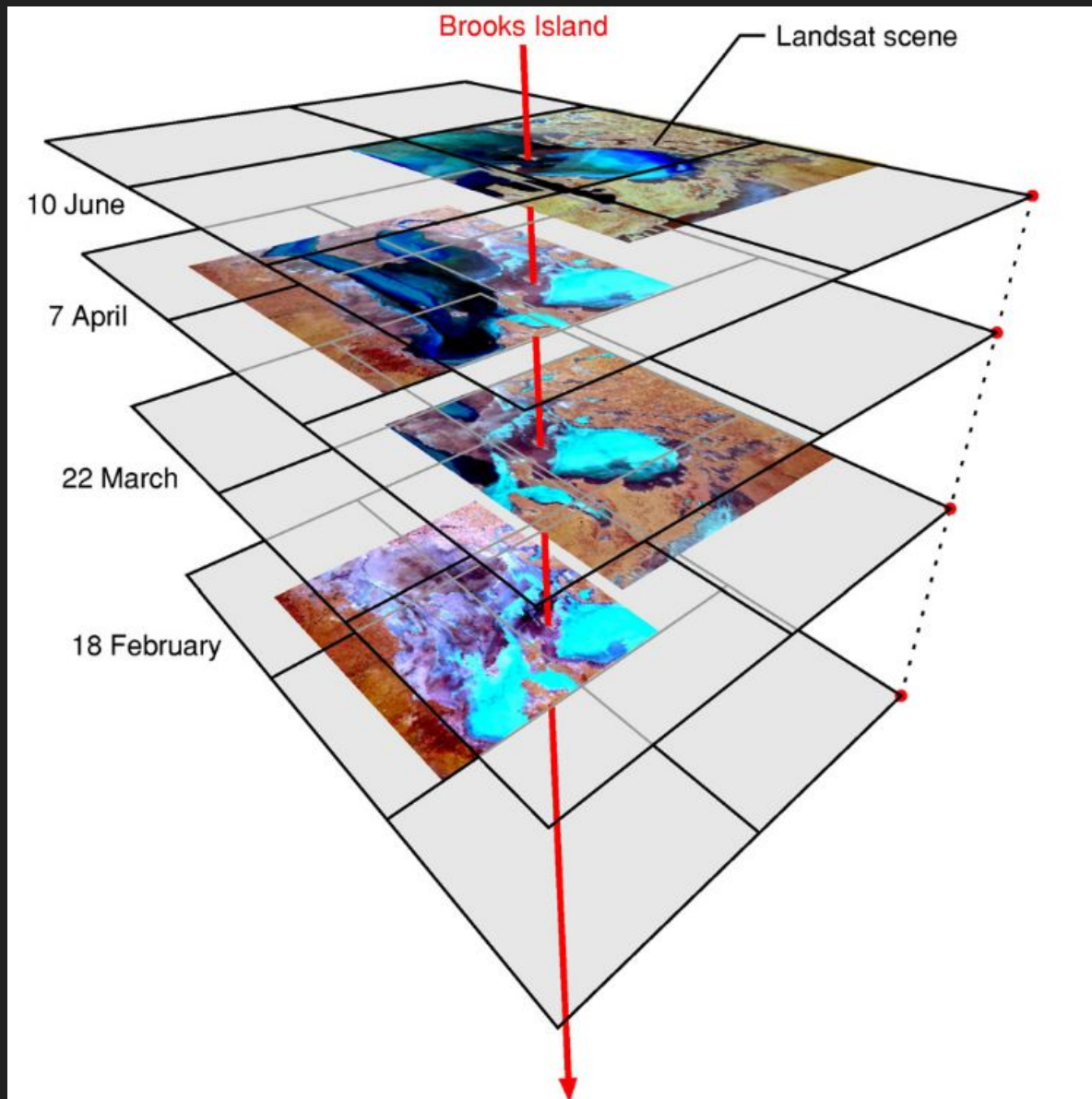












Open Data Cube (ODC)

Remote Sensing of Environment 202 (2017) 276–292



Contents lists available at ScienceDirect

Remote Sensing of Environment

journal homepage: www.elsevier.com/locate/rse



The Australian Geoscience Data Cube — Foundations and lessons learned



Adam Lewis^{a,*}, Simon Oliver^a, Leo Lymburner^a, Ben Evans^b, Lesley Wyborn^b, Norman Mueller^a, Gregory Raevksi^a, Jeremy Hooke^a, Rob Woodcock^c, Joshua Sixsmith^a, Wenjun Wu^a, Peter Tan^a, Fuqin Li^a, Brian Killough^d, Stuart Minchin^a, Dale Roberts^{a,b}, Damien Ayers^a, Biswajit Bala^a, John Dwyer^e, Arnold Dekker^c, Trevor Dhu^a, Andrew Hicks^a, Alex Ip^a, Matt Purss^a, Clare Richards^b, Stephen Sagar^a, Claire Trenham^b, Peter Wang^c, Lan-Wei Wang^a

^a Geoscience Australia, GPO Box 378, Canberra, ACT 2601, Australia

^b Australian National University, Canberra, ACT 2601, Australia

^c Commonwealth Scientific Industrial Research Organisation (CSIRO), GPO Box 1700, Canberra, ACT 2600, Australia

^d NASA Langley Research Center, 1 Nasa Dr, Hampton, VA 23666, United States

^e U.S. Geological Survey Earth Resources Observation and Science (EROS) Center, 47914 252nd Street, Sioux Falls, SD 57198, United States

ARTICLE INFO

Article history:

Received 27 July 2016

Received in revised form 7 February 2017

Accepted 12 March 2017

Available online 12 April 2017

Keywords:

Landsat

Time-series

Big data

Data cube

High performance computing

High performance data

Collection management

Geometric correction

Pixel quality

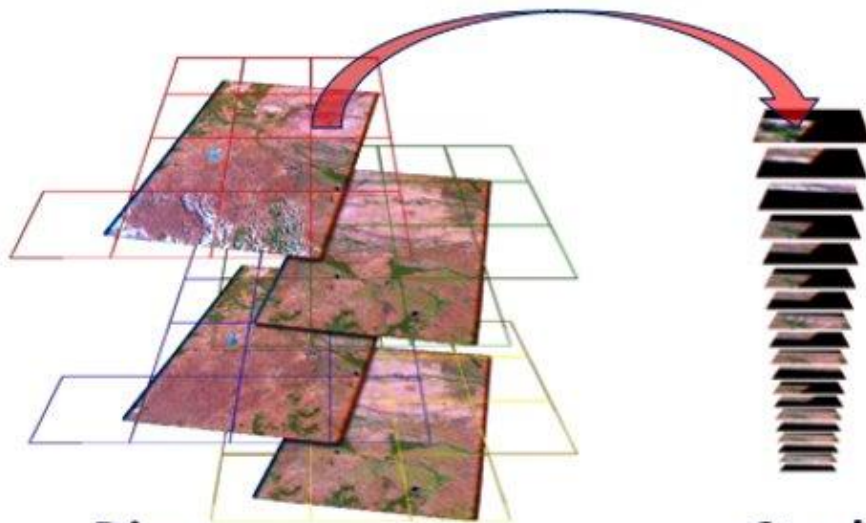
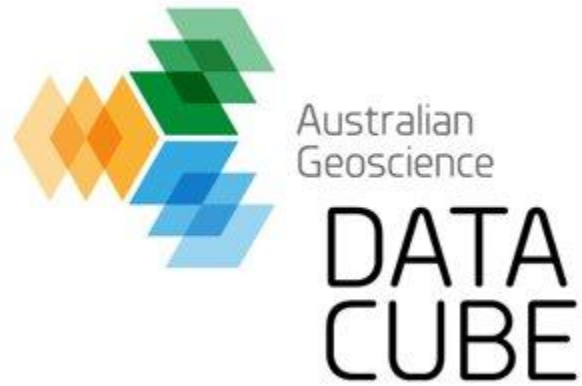
Australian Geoscience Data Cube

ABSTRACT

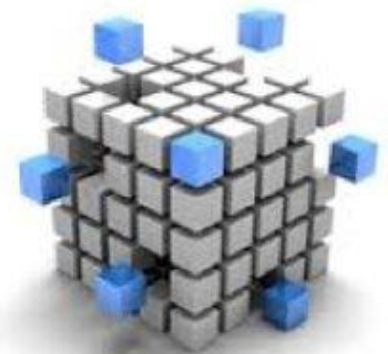
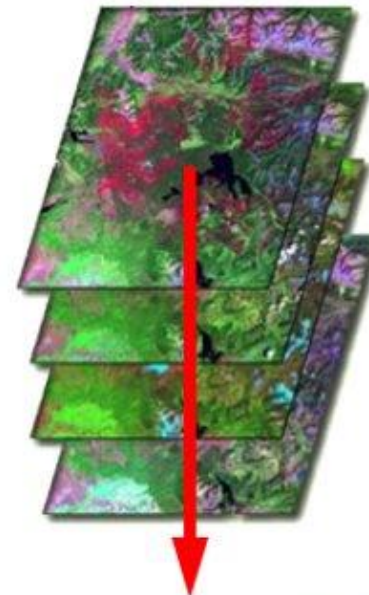
The Australian Geoscience Data Cube (AGDC) aims to realise the full potential of Earth observation data holdings by addressing the Big Data challenges of volume, velocity, and variety that otherwise limit the usefulness of Earth observation data. There have been several iterations and AGDC version 2 is a major advance on previous work. The foundations and core components of the AGDC are: (1) data preparation, including geometric and radiometric corrections to Earth observation data to produce standardised surface reflectance measurements that support time-series analysis, and collection management systems which track the provenance of each Data Cube product and formalise re-processing decisions; (2) the software environment used to manage and interact with the data; and (3) the supporting high performance computing environment provided by the Australian National Computational Infrastructure (NCI).

A growing number of examples demonstrate that our data cube approach allows analysts to extract rich new information from Earth observation time series, including through new methods that draw on the full spatial and temporal coverage of the Earth observation archives. To enable easy-uptake of the AGDC, and to facilitate future cooperative development, our code is developed under an open-source, Apache License, Version 2.0. This open-source approach is enabling other organisations, including the Committee on Earth Observing Satellites (CEOS), to explore the use of similar data cubes in developing countries.

Crown Copyright © 2017 Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).



... Stack ...

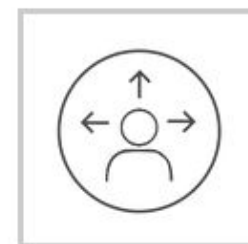
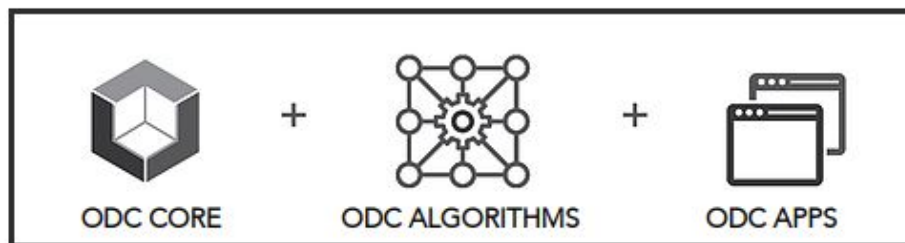
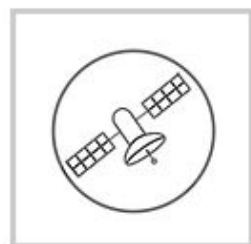


Open Data Cube (ODC)



Open Data Cube (ODC)

ODC ECOSYSTEM GEOSPATIAL DATA MANAGEMENT & ANALYSIS SOFTWARE



SATELLITE DATA

Examples:

- Landsat
- Sentinel
- MODIS

FLEXIBLE DEPLOYMENT

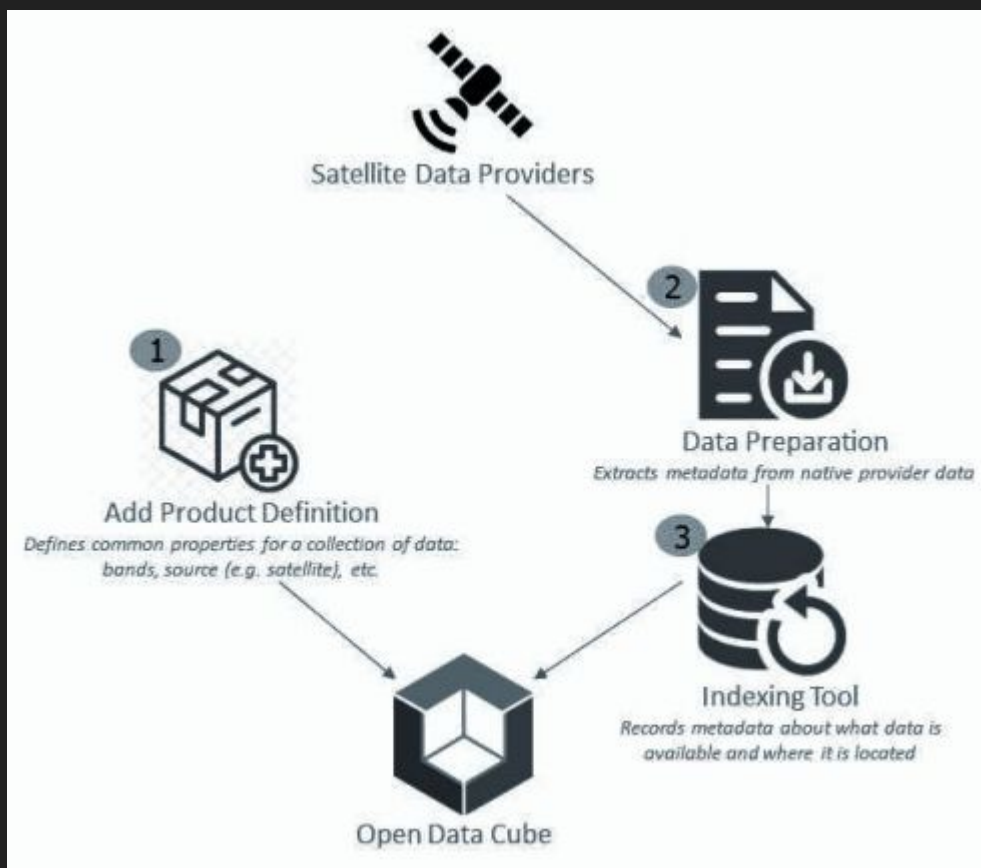
Depending on your application, the Open Data Cube can be deployed on HPC, Cloud, and local installations. Typical installations run on Linux, MacOS, and Windows.

INFORMED DECISIONS

Examples:

- Deforestation
- Water Quality
- Illegal Mining

Open Data Cube (ODC)



Google Earth Engine (GEE)



Pixel evaluated on visible map

Computation is highly reduced by sub-sampling the data

GEE does not store image data as a data cube but provides cube-like operations, such as reduction over space and time.

GDAL Cubes



Article

On-Demand Processing of Data Cubes from Satellite Image Collections with the gdalcubes Library

Marius Appel ^{*} and Edzer Pebesma

Institute for Geoinformatics, University of Münster, Heisenbergstraße 2, 48149 Münster, Germany

* Correspondence: marius.appel@uni-muenster.de

Received: 29 May 2019; Accepted: 26 June 2019; Published: 28 June 2019



Abstract: Earth observation data cubes are increasingly used as a data structure to make large collections of satellite images easily accessible to scientists. They hide complexities in the data such that data users can concentrate on the analysis rather than on data management. However, the construction of data cubes is not trivial and involves decisions that must be taken with regard to any particular analyses. This paper proposes on-demand data cubes, which are constructed on the fly when data users process the data. We introduce the open-source C++ library and R package gdalcubes for the construction and processing of on-demand data cubes from satellite image collections, and show how it supports interactive method development workflows where data users can initially try methods on small subsamples before running analyses on high resolution and/or large areas. Two study cases, one on processing Sentinel-2 time series and the other on combining vegetation, land surface temperature, and precipitation data, demonstrate and evaluate this implementation. While results suggest that on-demand data cubes implemented in gdalcubes support interactivity and allow for combining multiple data products, the speed-up effect also strongly depends on how original data products are organized. The potential for cloud deployment is discussed.

Keywords: earth observations; satellite imagery; R; data cubes; Sentinel-2

GDAL Cubes

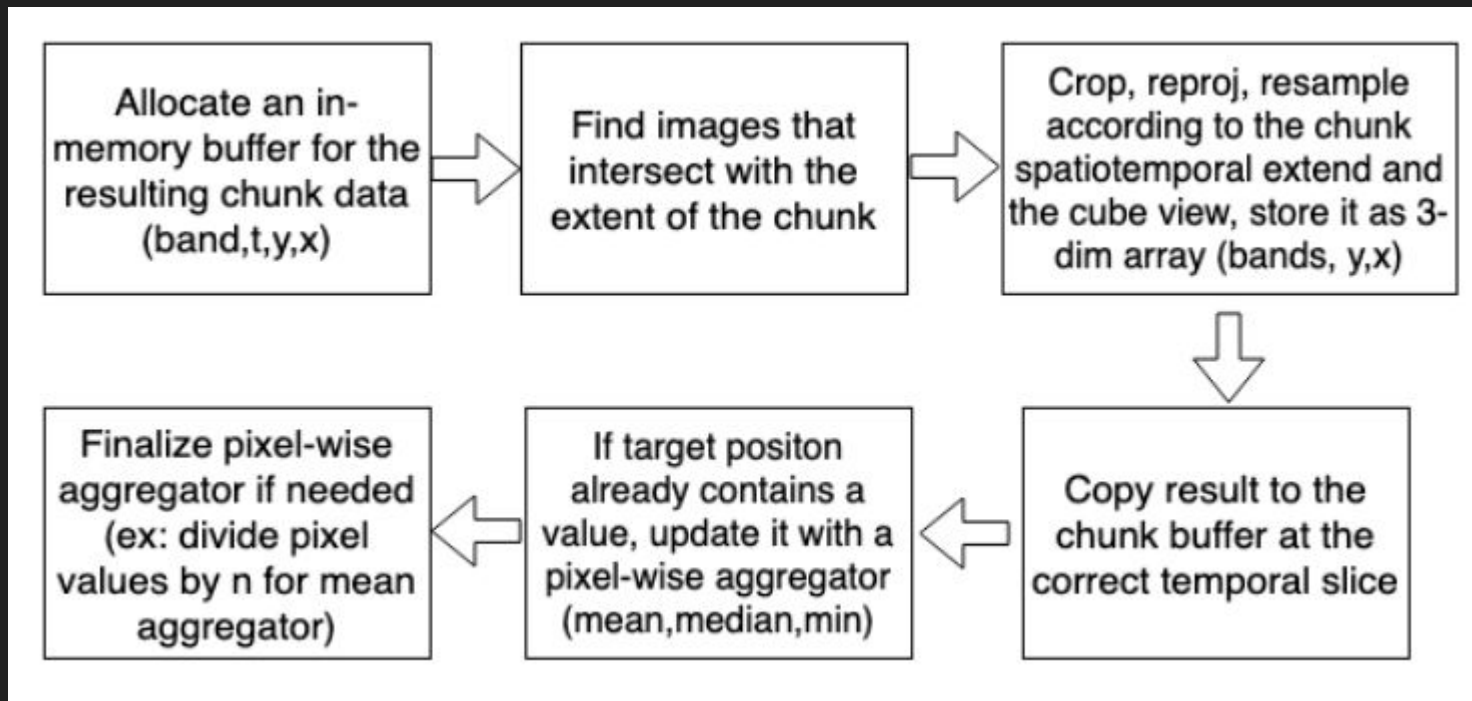


Definition 1. *A regular, dense raster data cube is a four-dimensional array with dimensions x (longitude or easting), y (latitude or northing), time, and bands with the following properties:*

- (i) Spatial dimensions refer to a single spatial reference system (SRS);*
- (ii) Cells of a data cube have a constant spatial size (with regard to the cube's SRS);*
- (iii) The spatial reference is defined by a simple offset and the cell size per axis, i.e., the cube axes are aligned with the SRS axes;*
- (iv) Cells of a data cube have a constant temporal duration, defined by an integer number and a date or time unit (years, months, days, hours, minutes, or seconds);*
- (v) The temporal reference is defined by a simple start date/time and the temporal duration of cells;*
- (vi) For every combination of dimensions, a cell has a single, scalar (real) attribute value.*

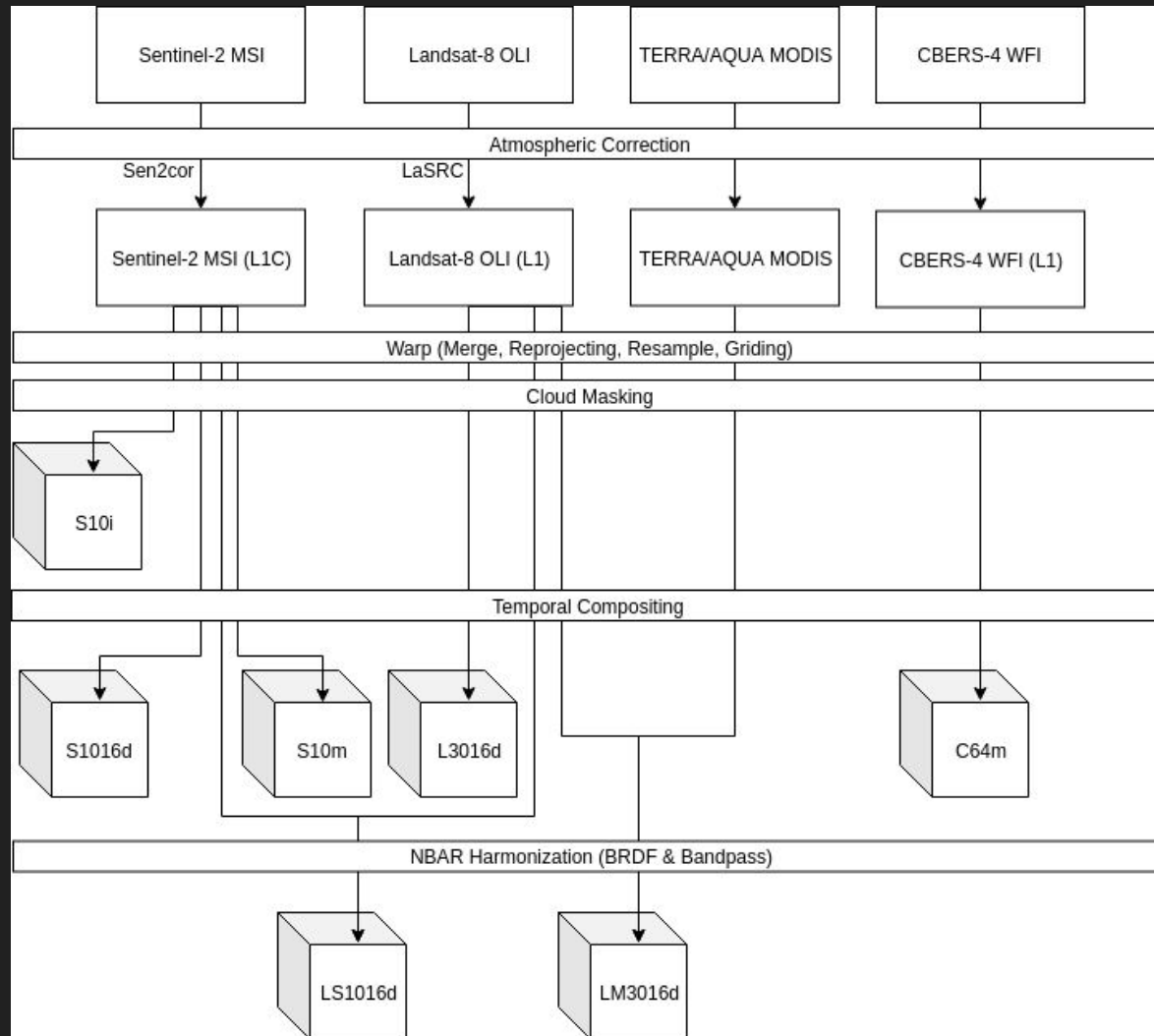
Definition 2. *An image collection is a set of n images, where images contain m variables or spectral bands. Band data from one image share a common spatial footprint, acquisition date/time, and spatial reference system but may have different pixel sizes. Technically, the data of bands may come from one or more files, depending on the organization of a particular data product.*

GDAL Cubes

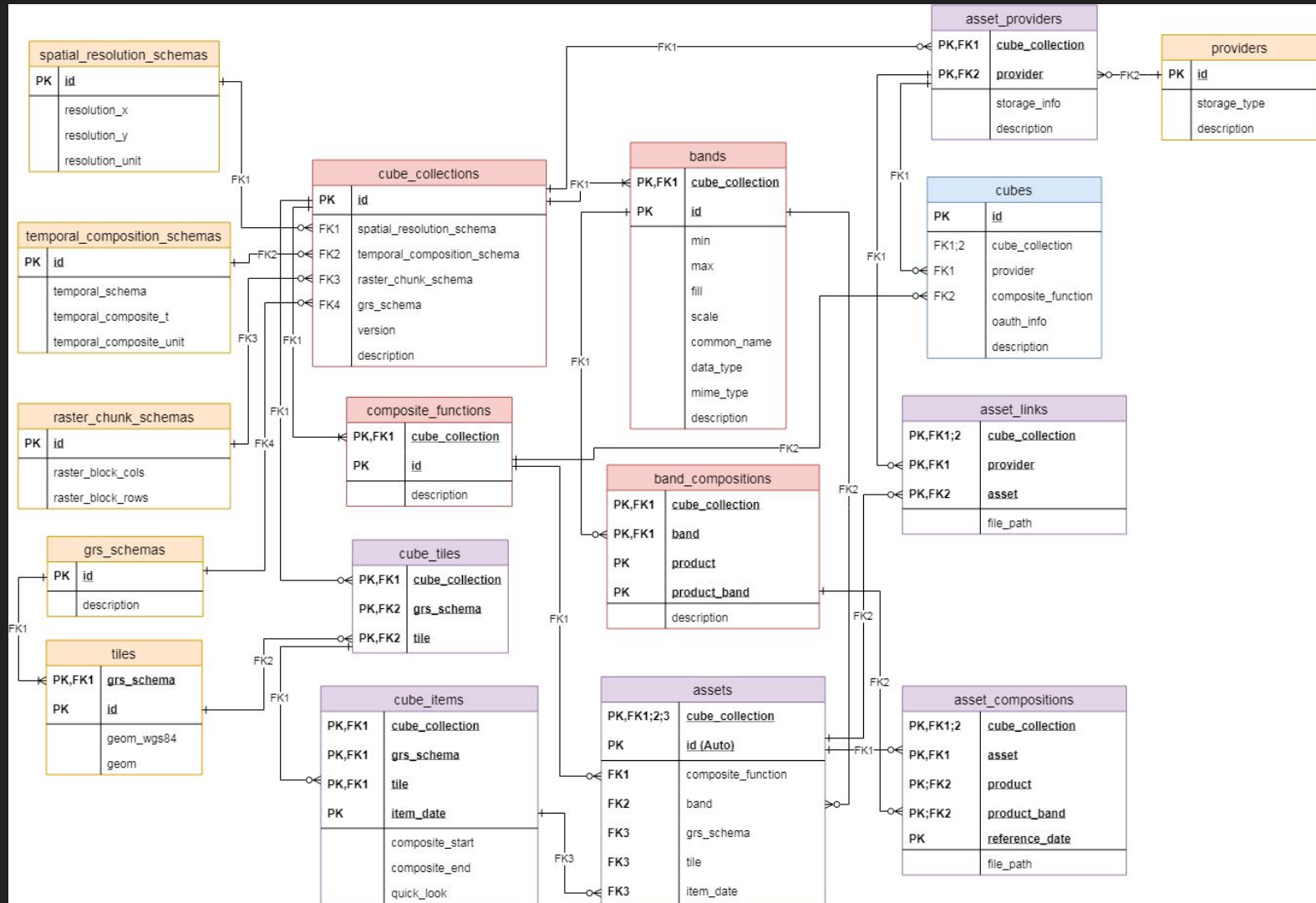


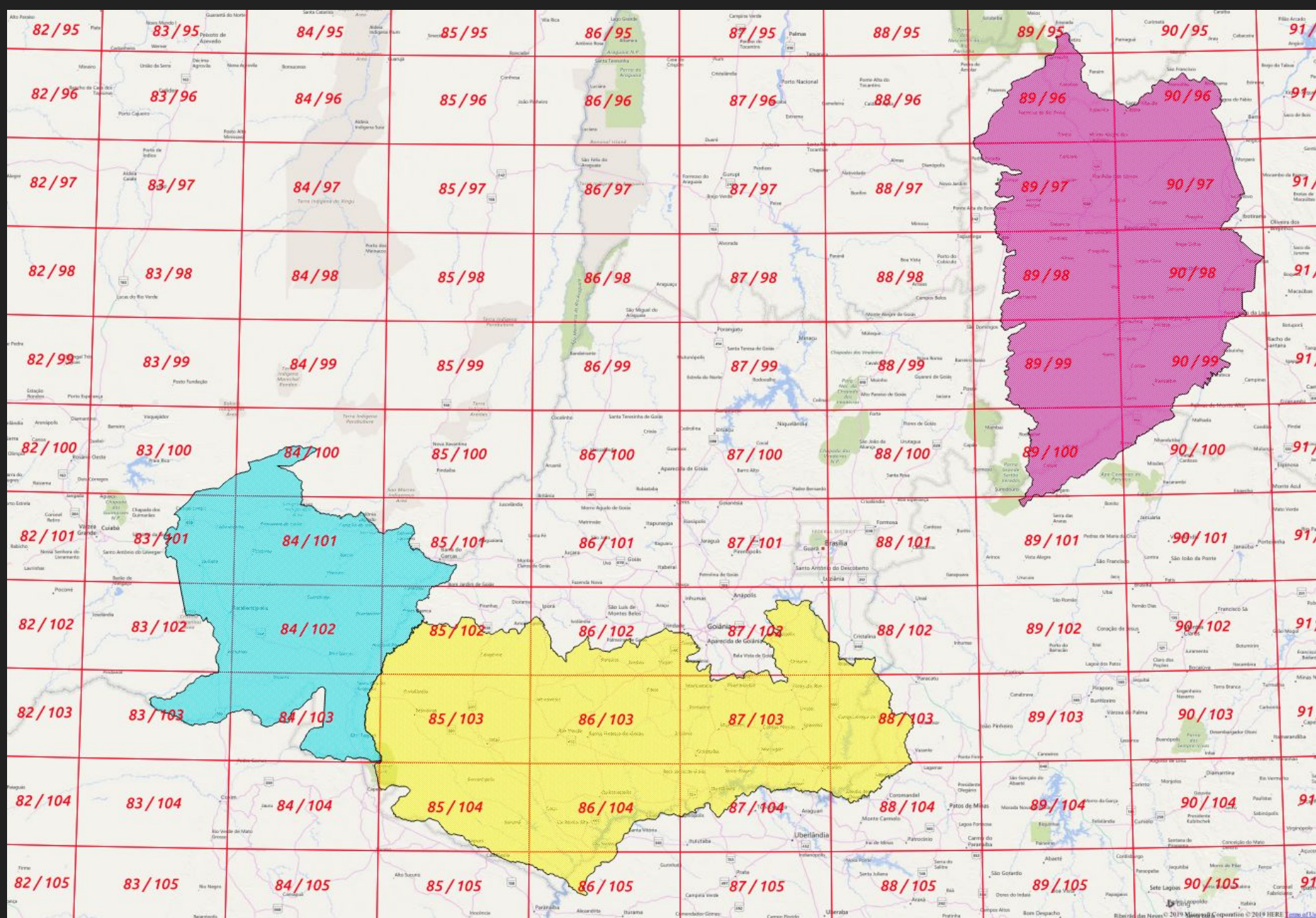
GDAL Cubes

Brazil Data Cube (BDC)

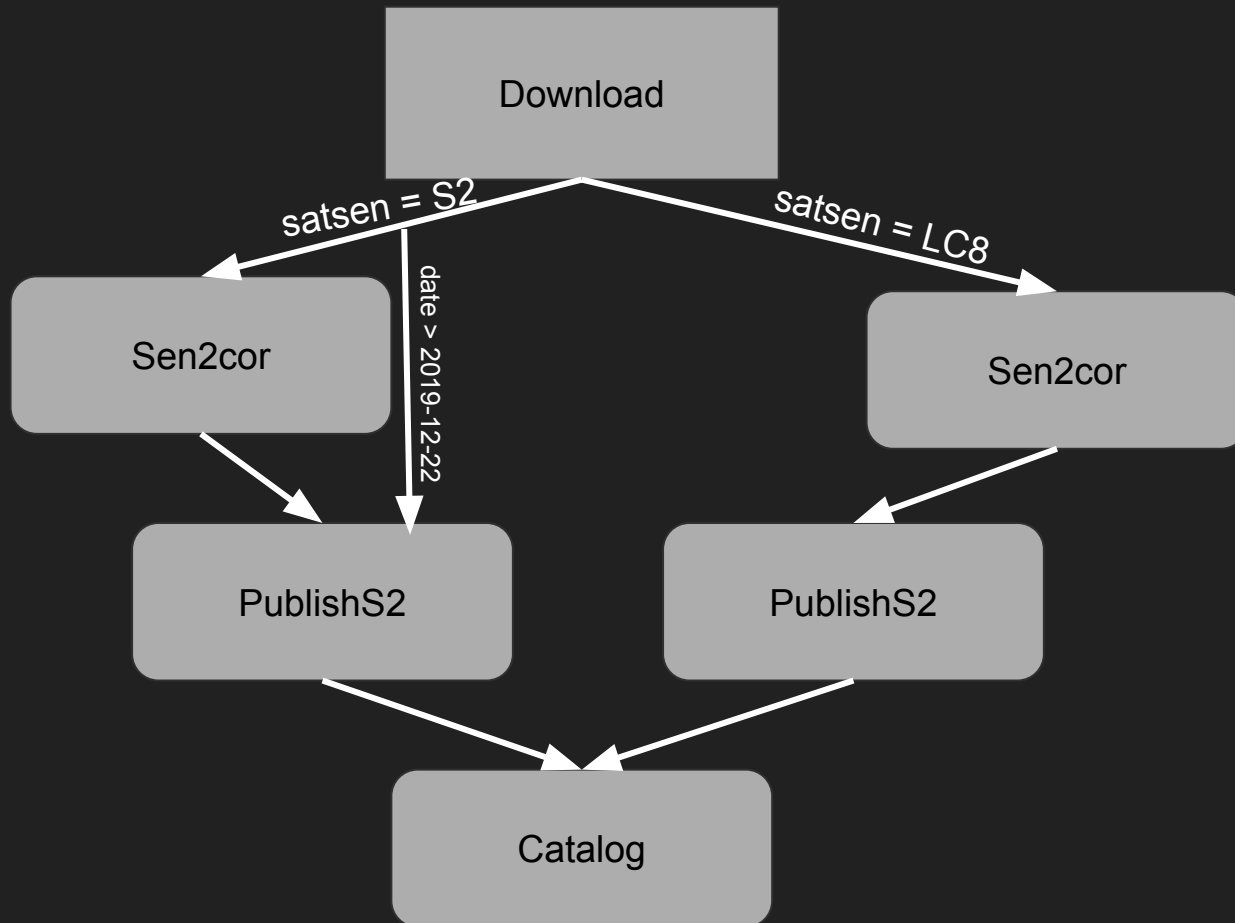


Brazil Data Cube (BDC)

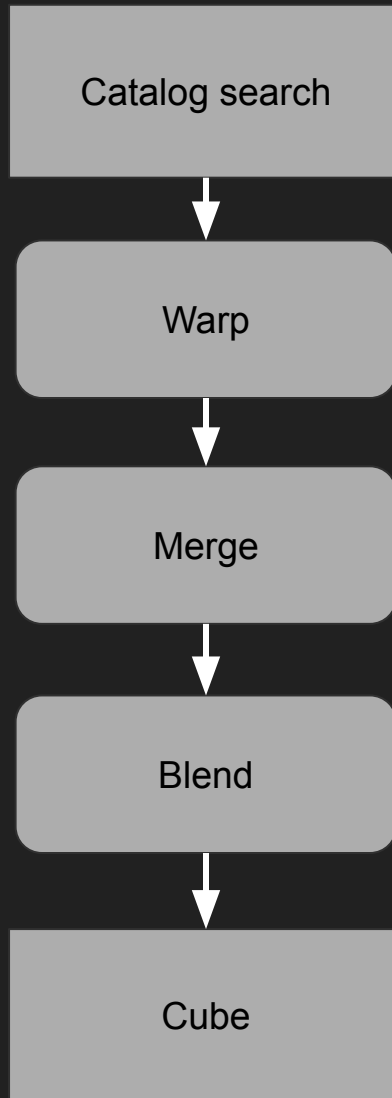




Brazil Data Cube (BDC): Radcor



Brazil Data Cube (BDC): Data Storm





<https://github.com/brazil-data-cube/bdc-scripts>

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

No description, website, or topics provided.

Edit

Manage topics

106 commits

2 branches

0 releases

3 contributors

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

jefersonarcanjo Delete aea_AWFI.prj ...

Latest commit 2e0a01f 3 days ago

ds_config	Delete aea_AWFI.prj	3 days ago
ds_cubesearch	changed permissions	10 days ago
ds_executive	changed permissions	10 days ago
ds_include	Initial version scripts for BDC System	4 months ago
ds_maestro	updates	3 months ago
ds_soloist	fix_merge	13 days ago
ds_source	updated sen2cor essentials	5 days ago
ds_utils	solving merge conflits	12 days ago
rc_espa-science	changed permissions	10 days ago
rc_maestro	updated restart, reset, logs	4 days ago
rc_sen2cor	updated sen2cor essentials	5 days ago
.gitignore	updated sen2cor essentials	5 days ago
README.md	changed permissions	10 days ago
bdc_compose.yml	add compose files via upload	12 days ago
bdc_swarm.yml	add compose files via upload	12 days ago
build.sh	changed permissions	10 days ago

BDC-Scripts - Brazil Data Cube

Infrastructure for creating brazil data cube environment in a local server machine

Structure

- `ds_config` : Definitions and system configs (Reference Grids, Spetral bands mapping, etc)
- `ds_cubesearch` : Web interface to view datacubes, download products, etc.
- `ds_executive` : Web interface for management of datacubes (definition, creating, execution)
- `ds_include` : Include files needed by the system
- `ds_maestro` : Docker and components that manage the cube generation system (manage queues, tasks)
- `ds_opensearch` : API opensearch to the original files repository
- `ds_soloist` : Docker and components that execute datacube tasks
- `ds_source` : Python source codes
- `ds_utils` : Python API for accessing data cube and general tools

Requirements

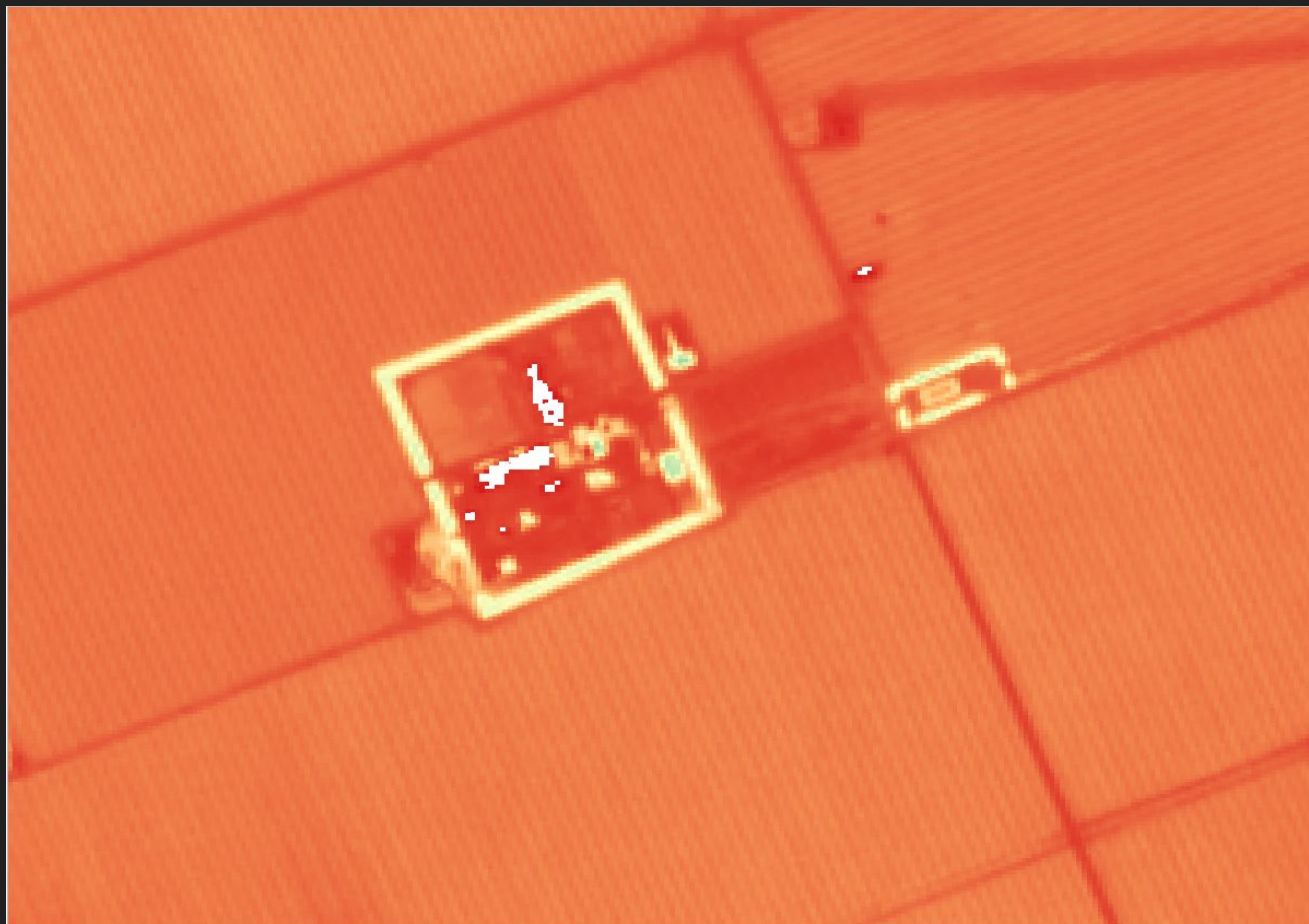
Make sure you have the following libraries and tools installed:

- `Python 3`
- `Docker` Docker version 18.06.1 or later
- `Docker Compose` docker-compose version 1.22.0 or later

Installation

Running with docker

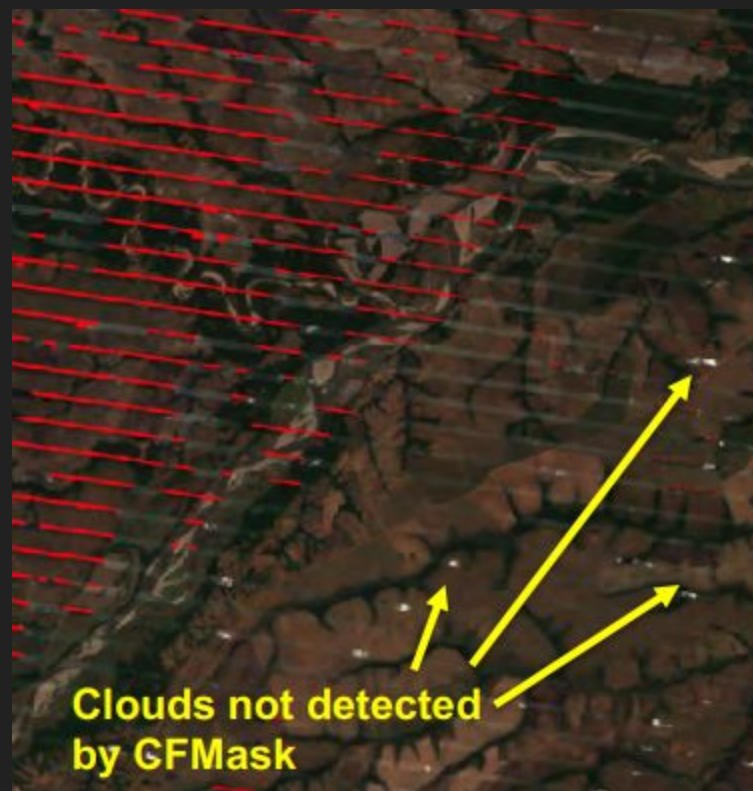
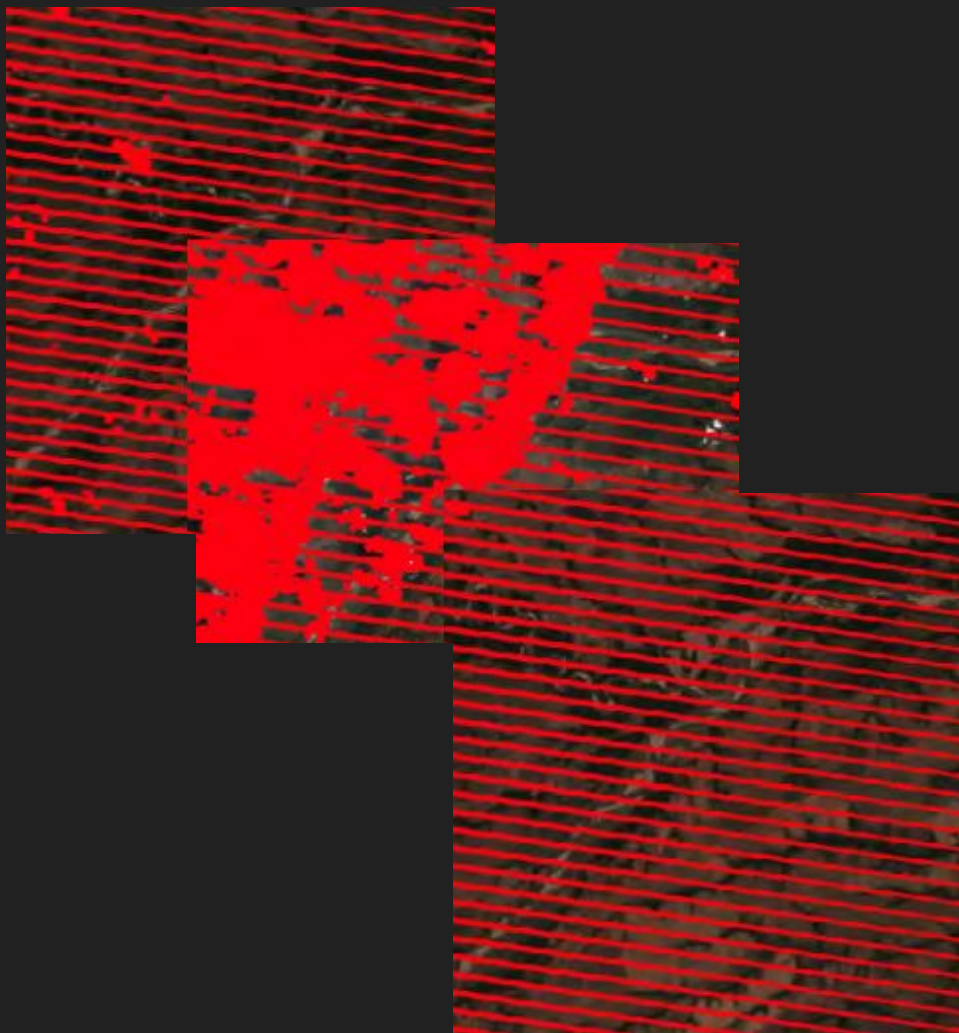
Cloud mask removing



Register



Cloud Mask



Cloud Mask



Cloud Shadow Mask

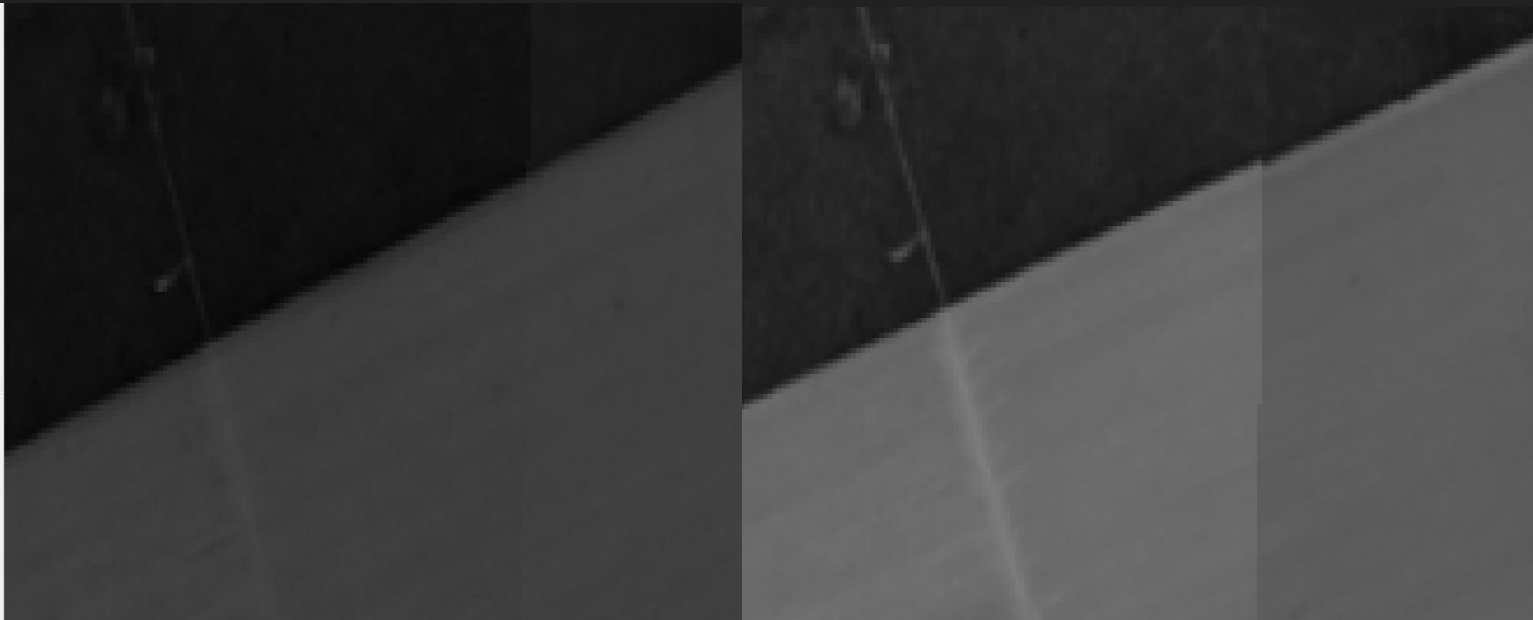


Sentinel Overlap



Sentinel Overlap

☐ S10m_089098_2018-09-01_coastal_STACK
☒ S10m_089098_2018-09-01_blue_STACK
☐ S10m_089098_2018-09-01_green_STACK
☒ S10m_089098_2018-09-01_red_STACK



Data Provenance